Technical evaluation of the CrowdPrisma TextEngine

February 7, 2025

1 Overview

This technical report aims to benchmark CrowdPrisma against three types of topic modelling approaches:

- traditional probabilistic ones LDA and NMF
- a newer one which is based on sentence embeddings from transformers and advanced hierarchical clustering BERTopic
- and a typical LLM based one that relies on prompt engineering best practices and zero shot learning.

The report is structured as follows:

- **CrowdPrisma background**: Provides some necessary background on the origin of CrowdPrisma as a product.
- **TextEngine overview**: Gives a high-level understanding of CrowdPrisma's TextEngine and it's three core parts.
- **Topic modelling - history & evaluation challenges**: Summarizes the history of topic modelling and the challenges related to the evaluation of these models.
- **Dataset and methodology**: Describes the dataset used in this evaluation report, along with the approach taken to compare the different topic modelling pipelines.
- **Results**: Overview of the results with key figures and tables.
- **Conclusions**: Provides key takeaways from this report, especially regarding the performance of CrowdPrisma against other methods.

We recommend the technical readers to take a look at the accompanying notebook, which contains all the experiments, details and code for this report.

2 CrowdPrisma background

CrowdPrisma was originally developed for policy research where qualitative surveys are often used to gather input from industry participants or the public. These surveys usually results in hundreds or thousands of pages of free-text responses, often written in multiple languages. Furthermore, the text data is usually mixed with non-text answers (i.e. numerical, categorical, multiple-choice, date). We needed a system that can automatically extract relevant topics from complex, technical text and can do this in a verifiable way. That is, we had to make sure our topic assignments can be pinned to certain sections of responses so that the end-user can check them and ultimately learn to trust the system.

With nearly two decades of experience (between the founders of CrowdPrisma) in Machine Learning

(ML) and Natural Language Processing (NLP), we started this journey back in 2022, before Large Language Models became mainstream. We built the first MVP of CrowdPrisma based on the technology that is powering BERTopic but we quickly realized that it is not powerful and precise enough for our users. So we welcomed ChatGPT and its competitors with open arms and rewrote the whole platform on top of LLMs. However, we soon found out that building a robust and verifiable topic modelling system on top of LLMs is not trivial.

Analysing surveys with hundreds of pages of free-text, poses unique challenges that cannot be solved by simply zero, one or even few-shot prompting LLMs: - When presented with a large document with many responses, LLMs cannot reliably tell where one response starts and another ends. As a result, LLMs blend and mix them together when asked for quotes. - LLMs cannot reliably and accurately quantify topics and themes with few-shot prompting within responses even when explicitly asked to correct themselves multiple times.

Several papers demonstrated that these challenges can only be overcome by more complex LLM based pipelines which consist of at least an order of magnitude more LLM calls, error correction steps, information aggregation and cross-checking steps. Based on this and our own research we developed CrowdPrisma's TextEngine, which is a state-of-the-art topic-modelling pipeline built on the latest LLMs and MLOps tooling.

3 TextEngine overview

The CrowdPrisma TextEngine consists of 3 main parts: topic extraction, theme creation and topic assignment. Each has a number of unique features, which when combined together make the whole TextEngine extremely powerful.

1. **Topic extraction**: For each text column in the dataset we automatically discover coherent and recurring topics. This part of the pipeline is
   - **Robust**: we perform topic extraction several times (each time on a previously unseen sample of the corpus) to minimize the chance of the LLM overlooking (or forgetting to mention) certain smaller topics and ultimately to achieve a robust set of final topics.
   - **Universally English**: our TextEngine will read and understand any language but it will always map the mentioned topics to English. This is by design: it ensures that we can keep accurate topic counts across multiple languages. This is crucial, because all of our clients are based in the US & EU working in English.
2. **Theme creation**: Once we found topics, we deduplicate them and form themes from them. Theme creation is
   - **Non-redundant and uniform**: deduplication and topic cleaning is key to achieve a final set of non-overlapping topics that have uniformly formatted names and descriptions.
   - **Hierarchical**: we group the discovered topics into higher level themes to aid the understanding of large, complex corpora. This allows researchers to reason about their dataset both at a high level (themes) and more granularly (topics).
   - **Iterative**: instead of doing this in a single call, we deliberately do the theme creation iteratively, asking the model at each turn to revise its previous list of themes and only add new topics to them if needed.
3. **Topic assignment**: In this stage, we assign each text response individually, avoiding quality issues that arise from batching multiple assignment requests into a single API call. Our assignment is

- **Verifiable**: for every topic assignment we extract a supporting quote. Therefore each topic assignment is easily verifiable by the end user in our Prisma Dashboard.
- **Natively multi-label**: to reflect the nuanced and complex nature of human language we allow our pipeline to assign each response to multiple topics (if needed). With multiple assignments the extracted supporting quotes can (and often) overlap, exactly as we'd see from a human analyst making sense of written text.

On top of the above features, as a whole, the pipeline is:

- **Resilient**: fault-tolerant to API limits & timeouts, natively guarded against malformed responses (that regularly break large LLM pipelines).
- **Highly configurable**: the behaviour and execution of the pipeline is governed by almost a hundred parameters which can be set and saved before every run (or re-run).
- **Prioritizes quality over cost or runtime**: this is evident in all parts of the pipeline: we extract topics multiple times, we form themes iteratively and we assign each response separately and not in batches. All of these increase cost and runtime, but also the quality of our results, which is what matters to our clients.
- **Linked to Prisma Dashboard**: the output of the TextEngine is turned into a highly interactive and infinitely filterable dashboard that allows the user to contrast and compare any subgroups. **See this video for further details.**

4 Topic modelling

4.1 Short history

Topic modelling is an unsupervised machine learning task that has received relatively limited attention by the AI community. The classic, probabilistic algorithm to attack the problem is called Latent Dirichlet Allocation by Blei et al. It works by first creating a `document x word` matrix and then probabilistically modelling each document as a bag of words, looking for characteristic words that define topics across multiple documents. The other classic (non-probabilistic) algorithm is Non-negative Matrix Factorization which again treats each document (or piece of text) as a bag of words. There are threads of research that actually connect these two algorithms.

The obvious problem with representing documents as a collection of words is that you lose a tremendous amount of semantic information. For example "Apple" means something very different in a stock market news and in a novel describing a market scene. Furthermore, you also miss out on a huge amount of complex information that is encoded in the text on levels higher than words, i.e. groups of words (e.g.: United States of America), sentences and paragraphs.

But the biggest problem with these algorithms that start from a `document x word` matrix is that said matrix consist of counts (which is why it lends itself to be modelled by a Dirichlet distribution in the first place). So each word is simply represented as a count vector across a collection of documents, allowing for the model to decipher co-occurrence patterns between different words and thereby deduct "topics" or rather groups of words that occur together in the corpus. But how would you interpret and make sense of these "topics"? They are nothing more than a list of words. Therefore (as we'll see later in our report) it is extremely challenging to describe these "topics" in a semantically-meaningful way and reason about them.

To alleviate these problems, the Natural Language Processing part of the wider machine learning community first started representing each word not as simple count vector in a `document x word`

matrix, but as a real vector which is *learned* from a huge corpus of text by trying to predict the words before and after the one in the middle. Doing so, encodes a lot more of the semantic richness we discussed above and leads to word vectors that are way more useful to work with in downstream tasks. The seminal paper for this area of research is the Word2Vec one from Facebook.

Later, this idea was expanded to get representations for not only single words but entire sentences or documents (see Doc2Vec). Then Google released the seminal Attention Is All You Need paper which invented the Transformer architecture of deep neural networks (the archetype of all modern LLMs). The following year they also released BERT a Bidirectional Transformer that was trained on a (by the standards of those days) huge corpus and represented a giant leap in NLP. Before that, most NLP subtasks (Entity Recognition, Knowledge Extraction, Sentiment Analysis, Summarization, Question Answering, Text Generation, etc), used a different model architectures to train on their specialized datasets. But after BERT (in a reverse Babel moment) these disparate fields could coalesce and work from the same model (BERT) by fine-tuning it to their use-cases. Sentence-BERT was also instrumental in this change as it allowed anyone to quickly get semantically meaningful and comparable vectors for any sentence that came from BERT like transformers.

In the context of topic modelling, these advances allowed researchers to represent documents in a rich multidimensional space that was learned to capture semantic meaning, then use clustering algorithms on these vector representations to find pieces of text that are similar (semantically) and therefore should be considered together as a single topic. This is the principal idea behind BERTopic, which has become a very successful open-source topic modelling and text exploration package.

Finally, in the last 2 years, affordable LLMs like GPT3.5 and GPT4 allowed researchers to leverage the incredible language understanding of these giant models and ask them directly to find recurring and relevant topics in a set of texts. TopicGPT was one of the first papers to show the potential of this approach and outline some key concepts in how to setup an LLM based topic modelling pipeline. Recent research has showed that LLM based topic modelling can rival specialized supervised models in this field, which is really a testament to the power of LLMs. Despite their many limitations (confident hallucination, preferential remembering of later inputs, etc), they do encode an unprecedented amount of linguistic knowledge and when prompted and guard-railed correctly they can produce stellar performance on completely unseen data.

4.2 Problems in topic model evaluation

Topic modelling consists of two subtasks: topic extraction (or discovery) and topic assignment (or classification). The evaluation of the former is a nightmare, while the evaluation of the latter is much more straightforward (like all supervised ML tasks).

The extraction part of topic modelling is an unsupervised NLP task. Therefore, evaluating models in this space is notoriously difficult. There are three types of evaluation methods:

1. **Coherence scores** like UMass, NPMI and CV measure the consistency of words in a given topic to evaluate the interpretability and meaningfulness of a topic by computing a level of semantic similarity among words that are included in the topic.
2. **Model based scores** such as perplexity and likelihood are calculated on held-out data to assess how well the model generalizes to new unseen data. Lower perplexity and higher likelihood scores indicate better generalization.
3. **Human evaluation** relies on the judgement of humans to access the interoperability and

quality of topics and ensure they are clear, understandable, don't repeat, overlap or miss crucial parts of the corpus.

There are unique challenges posed by all three:

1. Coherence scores are cheap to calculate and give a single numerical value which is tempting as it makes ranking of models trivial. However all of these scores rely on a reference corpus and evaluate topic quality based on a word level co-occurrence probabilities, which makes their assessment really crude. This is why, these scores often don't align at all with human judgement. In fact, these scores are so misaligned with human judgement of topic quality that recently researchers have proposed to instead outsource human evaluation to LLMs.
2. Model based scores are just as misaligned with human judgement, plus they are only really useful for evaluating a trained model's response on unseen data.
3. Human evaluation is slow, error prone and inherently fuzzy, which is why we often ask several judges to go through the topics. Nonetheless, it is still the best in terms of making sure the topics make sense and align with expectations.

As stated above, measuring the performance of the assignment/classification part of a topic modelling pipeline is straightforward. One can use any multi-class classification metric, but generally speaking weighted precision and recall scores or their combination (F1 score) is recommended.

5 Dataset and benchmarking methodology

5.1 Dateset

This report is using Bitex's customer service data for training LLM assistants. It has 26872 question/answer pairs from customer service interactions with associated labels for `category` and `intent`. There are 11 categories and 27 intents (shown in greater detail below in the notebook). For each intent we have roughly 1000 samples.

This dataset is very different from the ones we developed CrowdPrisma on. This customer service dataset represents a corpus of text that is very simplistic linguistically with a mean of 10 words per user query. However this level of complexity is typical of a lot of online dataset so we thought it will be challenging and illuminating to test CrowdPrisma on it.

Below are the categories and intents within the dataset

Category	Intent	Count
ACCOUNT	create_account	997
ACCOUNT	delete_account	995
ACCOUNT	edit_account	1000
ACCOUNT	recover_password	995
ACCOUNT	registration_problems	999
ACCOUNT	switch_account	1000
CANCEL	check_cancellation_fee	950
CONTACT	contact_customer_service	1000
CONTACT	contact_human_agent	999
DELIVERY	delivery_options	995
DELIVERY	delivery_period	999
FEEDBACK	complaint	1000

Category	Intent	Count
FEEDBACK	review	997
INVOICE	check_invoice	1000
INVOICE	get_invoice	999
ORDER	cancel_order	998
ORDER	change_order	997
ORDER	place_order	998
ORDER	track_order	995
PAYMENT	check_payment_methods	999
PAYMENT	payment_issue	999
REFUND	check_refund_policy	997
REFUND	get_refund	997
REFUND	track_refund	998
SHIPPING	change_shipping_address	973
SHIPPING	set_up_shipping_address	997
SUBSCRIPTION	newsletter_subscription	999

5.2 Methodology for comparing the topic modelling methods

We will compare CrowdPrisma's TextEngine to four models:

- **LDA**: classic, probabilistic, `document x word` matrix based topic modelling algorithm.
- **NMF**: classic, non-probabilistic, `document x word` matrix based topic modelling algorithm.
- **BERTopic**: modern, popular, transformer + clustering based topic modelling pipeline. BERTopic first represents documents as high-dimensional vectors (embeddings) that it calculates with Sentence-BERT. It then uses a non-linear dimensionality reduction technique (UMAP) to reduce the number of dimensions of the embeddings. Finally it discovers clusters in them using a hierarchical density based clustering algorithm (hdbscan). Finally it can extract keywords or ngrams from each cluster to represent them as a topic.
- **LLM based topic model (LLMTM)**: We use GPT4o and ask it to extract relevant topics from the corpus. We then also ask it to assign each response to one of the topics.

Since CrowdPrisma specializes in the understanding and analyzing of text responses within a survey (without any input or guidance from the user), both topic discovery and assignment are extremely important to us. Therefore in this report we will measure the performance of both subtasks as separately:

5.2.1 Topic discovery

- Run LDA, NMF, BERTopic, LLMTM on the entire 27k dataset.
  - For LDA, NMF and BERTopic we will set the number of topics to extract as the number of categories (11) and intents (27) respectively.
  - For LLMTM we will first ask for "11 customer service categories" then "27 customer intents". Separately, we will also run the LLMTM without any specific instructions.
  - Furthermore, we'll run two versions of each LLMTM model
    - one where the entire dataset is used for modelling, where we feed 5k batches at a time to the model for extraction and concatenate the extracted topics -> this gives

this model an unfair advantage as it has a go at the problem 6 times, leading to high recall (but lower precision).
    * one where we take a random 5k sample of the dataset and do topic extraction on that - this is much closer to what CrowdPrisma sees from the data.
  - CrowdPrisma will simply be ran with its default settings, without any additional help or input and hopefully it will recover the categories as themes and the intents as topics in one go. We simply pretend that this dataset is a column of a survey, and the column header is posing a question about customer categories and intents. That's the only input the system will get.
- Following the ideas of Rahimi et al, we will use a special LLM based verifier to link up topics with actual categories and intents (i.e. the extracted topics of these pipelines).
  - This method takes in a list of words (from LDA, NMF or BERTopic) or topic description (from LLMTM and CrowdPrisma) and tries to match it with a provided list of topic names (categories and intents in our case).
  - We use GPT4o for this.
  - We do this matching 5 times for each model and for each target, to mimic a 5 people human eval team.
- We collate the results and calculate how many categories and intents were discovered and missed by each method. We summarise the performance using weighted precision, recall and F1 scores.

5.2.2 Topic assignment

- Define a 3k (~11%) random set of the 27k dataset to test the topic assignment capabilities of all pipelines. We do this subsampling to speed up the eval cut the cost of the LLM based pipelines.
- For all pipelines we'll use their extracted intents (27 true labels) from the previous subtask. This naturally puts an upper bound on the achievable performance for each method. This is deliberate and reflects the real world application of topic modelling systems, where the two subtasks are often interlinked.
- LDA, NMF and BERTopic does the two subtasks (extraction & assignment) in one go, so they require no further compute at this stage.
- For LLMTM, we write a LLM based classifier called LLMA. Then run the test set queries on GPT4o-mini and assign them to the intents discovered by the LLMTM (full), LLMTM (5k) and LLMTM - no instruct (5k) models, one by one.
- For CrowdPrisma we run the standard assignment pipeline on the test set, but force it to use GPT4o-mini (unlike in production, where it can intelligently switch between LLMs).
  - To circumvent the problem of CrowdPrisma returning multiple assigned labels, we use the most common or first assigned label for each query.
  - We manually check a few assignments where CrowdPrisma has assigned multiple intents for a given query to see if they makes sense.
- As a baseline (or contrast) we'll also add a supervised model: we embed all queries using sentence transformers and then train a LightGBM classifier on the train set, then predict the test set.
- Finally, we calculate weighted precision, recall and F1 for all methods.

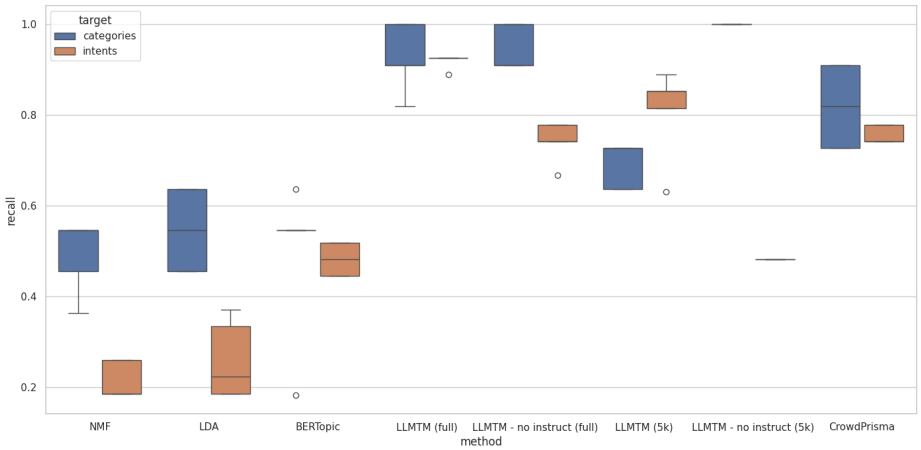
6 Results

6.1 Topic extraction / discovery

After running the LLM based topic verifier / matcher algorithm against all topic extraction models five times, we plotted the Weighted Recall, Precision and F1 scores of these methods. Note, the performance on Categories (11 true labels) and on Intents (27 true labels) are displayed in the same figure for all pipelines.

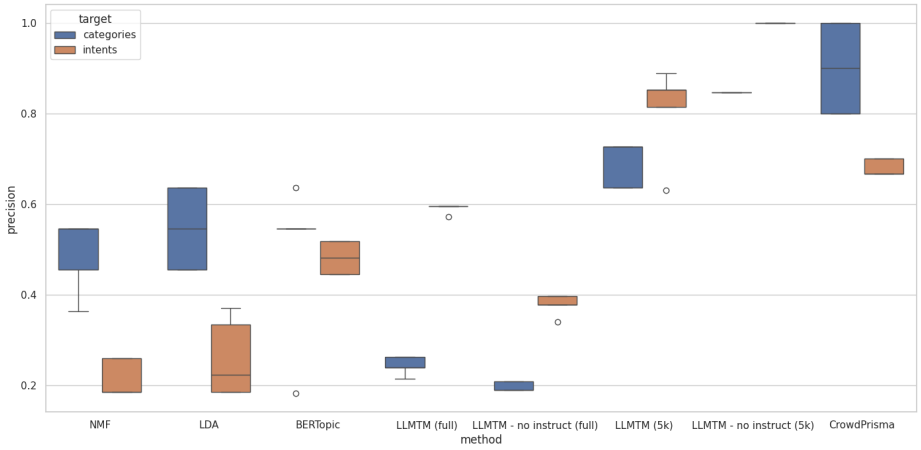
6.1.1 Recall

- There's a clear trend: newer methods find more and more of the topics (both categories and intents).
  - LDA and NMF are comparable, BERTopic is clearly better when it comes to finding intents (27 vs 11 categories).
- LLM based methods then represent another significant jump in recall. These methods almost always find ~80% of the topics, both in categories and intent.
  - The basic LLM based topic extractor, LLMTM performs really well, but it has two (unfair advantages): it runs 6 times (seeing a 5k batch in each iteration) and can form a superset of the extracted topics of each batch as its final result. Plus, it is told exactly what to look for: namely, "11 customer service categories" and "27 user intents".
  - When we remove either of these aids, its recall drops. The "LLMTM - no instruct" model find about 15% less intents, presumably because it is not told explicitly to return 27 intents. Furthermore the "LLMTM - no instruct (5k)" version (which does not get to read 6 5k batches but only one, and doesn't know how many categories and intents it needs to extract) gets all the categories right but only half the intents (almost 50% lower recall than LLMTM (full).
  - CrowdPrisma finds ~80% of the categories and intents without any input, while only reading 5 small samples of the data (so nowhere near as much as the LLMTM (full) sees).



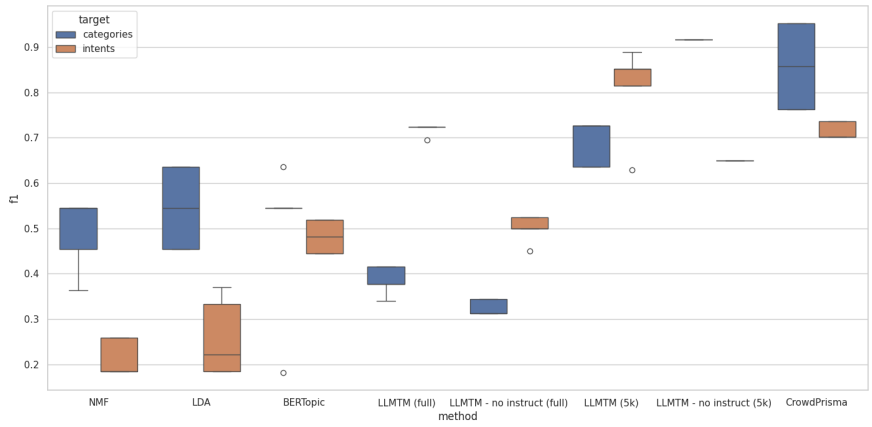
6.1.2 Precision

- A similar trend can be seen in precision. Older models have low precision, BERTopic medium precision, while LLM based systems have higher (or very high).
- Note: since the LLMTM (full) models have seen the data in 6 batches, and form a superset of the extracted topics of each batch, their list of topics is longer than needed. This is why they had high recall, and this is why they pay the price of for it here and get low precision.
- LLMTM (5k) models have higher precision, in some cases comparable to CrowdPrisma (the highest overall if we look at both categories and intents).



6.1.3 F1 score

- When we look at the combination of precision and recall (i.e. the F1 score) we see that CrowdPrisma has the highest overall score (if we are looking at categories and intents combined). This is impressive, since it wasn't told how many categories or intents to extract (or indeed that it should extract those at all).
- Furthermore, unlike all other methods, CrowdPrisma extracted these two targets (categories and intents) in one go, while forming a hierarchy from them automatically without any input (see the extracted themes and topics below).



6.1.4 Clarity and interpretation of topics

Another key consideration with topic modelling methods is the clarity or interpretability of their extracted topics. As we discussed in the history of topic modelling, this is an area that has seen dramatic improvement with LLMs. Below are some randomly selected topics from LDA, NMF, BERTopic and LLMTM.

Method	Example topic
LDA	phone, utell, acn, current, showing, free, tracking, removing, uhave, deleting, swap, add, changing, track, secondary, article, eta, change, number, order
NMF	locate, getting, accepted, checking, quick, account, want, delivery, order, options, look, methods, download, payment, invoices, check, bills, invoice, assistance, person
BERTopic	order,order number,number,order order,purchase order,purchase,status,number need,of order,cancel,eta,eta of,of,of purchase,the eta,number want,help,number help,cancel purchase,to cancel
LLMTM	address_update: Help with updating or modifying the delivery/shipping address.

As we can see, working with the first three requires a lot of mental effort, yet the topics described are still not necessarily clear and precise. That's the downside of representing topics a bag of words. Note, BERTopic now support using LLMs to generate topic descriptions instead of relying on bi-grams to represent a topic, but the results will never be great because the LLM is asked to work from the bi-grams or bag of words to start with.

Comparatively, the clarity we see with LLM based topics is refreshing. The description of LLMTM's topics are short, yet precise. However they can't match the output of CrowdPrisma, which not only extracted all the categories and intents automatically in one go (instead of two separate runs), but organised them into a neat hierarchy as shown below.

CrowdPrisma's automatically discovered categories and intents

- **Order Management**: Assistance with various aspects of order handling, including status updates, modifications, cancellations, tracking, and estimated time of arrival inquiries.
  - Order status: Request for information on the status of a refund or restitution.
  - Order modification: Assistance needed for changing an existing order.
  - Order cancellation: Requests related to canceling purchases.
  - Order ETA Inquiry: Requests for information on the estimated time of arrival for orders.
  - Order tracking assistance: Requests for help in tracking the status of product deliveries.
- **Refund Handling**: Inquiries and assistance regarding refund policies, status, and processes.
  - Refund policy: Inquiry regarding the money-back guarantee offered.
  - Refund inquiries: Requests for information about the status of refunds.
  - Refund process inquiries: Questions regarding the procedures and conditions for requesting refunds.

- **Account and Profile Support**: Assistance with issues related to account and profile management, including registration, personal information editing, and account closure.
  - Profile management: Request for assistance in editing personal profile information.
  - Registration issues: Requests for assistance with account registration problems.
  - Account management: Assistance needed for managing user accounts and premium features.
  - User profile creation: Questions related to the process of creating user profiles.
  - Account Closure: Requests for information on how to close accounts.
- **Delivery and Shipping Support**: Assistance with managing delivery addresses, shipping inquiries, and available delivery methods.
  - Delivery methods: Inquiry about available delivery options.
  - Delivery address management: Assistance needed for setting or modifying delivery addresses.
  - Shipping inquiries: Questions related to shipments and delivery locations.
- **Payment and Billing Inquiries**: Queries related to payment issues, available payment options, and understanding billing statements.
  - Payment Issues: Queries related to problems or troubles with payments.
  - Payment options: Inquiries about available payment methods and options.
  - Billing inquiries: Questions related to understanding or clarifying billing statements.
- **Service and Subscription Queries**: Inquiries related to managing subscriptions, termination charges, and freemium account features.
  - Subscription management: Assistance needed for managing newsletter subscriptions.
  - Termination charges inquiry: Request for information regarding charges associated with service termination.
  - Freemium account management: Inquiries related to managing information on freemium accounts.
- **Customer Support Information**: Requests for information on customer support hours and assistance in connecting with customer service representatives.
  - Customer support hours: Request for information on the hours of operation for customer assistance.
  - Customer Assistance: Requests for help in connecting with customer service representatives.
- **Policy and Fee Information**: Questions regarding the specifics of cancellation policies and various service-related fees.
  - Cancellation policies: Questions about charges or policies related to order cancellations.
  - Fee Inquiries: Questions regarding fees associated with services or accounts.
- **Feedback and Review Guidance**: Inquiries about the processes for submitting reviews and providing feedback to the company.
  - Review submission: Inquiries about how to leave reviews for services.
  - Feedback Process: Inquiries about how to provide feedback to the company.
- **Claims and Invoice Support**: Requests for assistance with filing claims and retrieving invoices.
  - Claims assistance: Requests for help in filing claims against the organization.
  - Invoice retrieval: Request for help in downloading or accessing invoices.

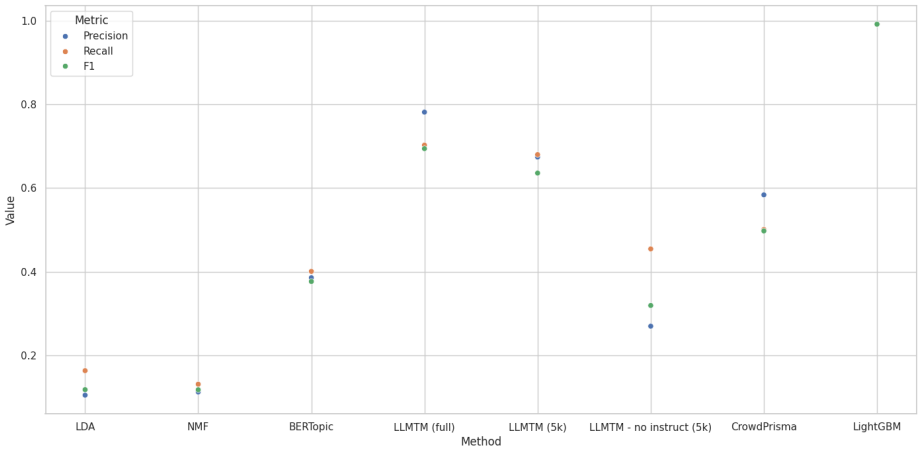
6.2 Topic assignment / classification

6.2.1 Overview

- To assess the topic assignment capabilities of these algorithms, we use Intents as target, since they are more granular (27 intents vs 11 categories).
- LDA, NMF and BERtopic does topic extraction and assignment at the same time.
- However, for LLM based methods we need to assign each query to an intent and effectively fire an API call to the LLM. To keep compute time and cost down, we select a random 3k (11%) subset of the data to test. The same set was used for all other methods.
- For LLMTM we used 3 versions to understand the effect of both subsampling (5k) and no instructions on the assignment performance.
- Then we calculated the weighted precision, recall and F1 scores of all methods.
- **Note**, for all methods, the assignment step is upper bound by the extraction step's performance. That is, if a method only found 10 intents of the 27, then it will never be able to get higher than 10/27 recall at the assignment stage.

6.2.2 Summary of results

- As shown in the figure (and table) below, we have a clear trend again. Older models are the worst, BERTopic is a lot better, while LLM based models represent another jump. The supervised model (LightGBM) is near perfect, which is not surprising on such a linguistically simple dataset.
- The three LLMTM models tell an interesting story, especially in comparison to CrowdPrisma:
  - LLMTM (full) has the same benefits as it had at extraction time: since it knew how many intents to fish out and it could have a go at the problem 6 times, it created a superset of topics. At assignment time, this comes in handy because it has identified all the 27 true intent labels. So unsurprisingly it achieves the highest recall of all unsupervised models.
  - Once we don't allow it to look at all the data 6 times, it extracts less intents and therefore at assignment time its performance is about 10-15% worse.
  - Once we also deny it the benefit of knowing what to extract (intents) it's suddenly operating in the same environment / setup as CrowdPrisma. However it performs significantly worse than CrowdPrisma. It's precision is 30% lower while it's F1 score is 20% lower.



Method	Precision	Recall	F1
LDA	0.104847	0.163333	0.11796
NMF	0.11245	0.131	0.11774
BERTopic	0.385474	0.400667	0.3765
LLMTM (full)	0.781172	0.702333	0.693944
LLMTM (5k)	0.67382	0.679667	0.635551
LLMTM - no instruct (5k)	0.269651	0.454333	0.319107
CrowdPrisma	0.583485	0.500667	0.497217
LightGBM	0.991433	0.991333	0.99134

6.2.3 Multi-label assignments

CrowdPrisma assigned 3% of the test set to two labels. Below are some examples. Arguably many of these could be argued to be about one of the two topics and CrowdPrisma rightly puts that as the first one. However, we can easily see, how with longer, more nuanced queries this feature of the CP TextEngine becomes invaluable.

- I want to send my feedback, I need assistance
  - Assigned intents: Feedback Process, Customer Assistance
- I have got to see how long refunds take, will you help me?
  - Assigned intents: Refund inquiries, Customer Assistance
- I do not know wha to do to check when will my product arrive
  - Assigned intents: Order ETA Inquiry, Order tracking assistance
- I do not know what to do to reset the key of my user profile
  - Assigned intents: Password reset assistance, Profile management

6.2.4 Verifiability

CrowdPrisma was also the only method that could explain why it assigned each response to a certain intent by extracting a supporting quote for the assignment (see example below). Again, with such short and simple queries, this isn't an issue, but with responses that span several pages, knowing the source of a topic assignment is key.

- I paid {{Currency Symbol}}{{Refund Amount}} for this product, assistance to receive a refund
  - Assigned intent: “Refund inquiries”
  - Supporting quote: “assistance to receive a refund”

7 Conclusions

- Topic modelling is an unsupervised ML task that consists of two subtasks (extraction and assignment). A pipeline’s performance on the first task puts an upper bound on it’s performance on the second subtask (and overall performance naturally).
- We evaluated CrowdPrisma against 4 different types of topic modelling methods with various setups (8 in total). Although CrowdPrisma proved to be the most precise and requiring the least input, which pipeline to choose depends a number of factors.
  - In settings where we don’t have labelled data or any background knowledge of the topics, CrowdPrisma provides the absolute best performance, in both stages of the pipeline. The authors suspect the difference between CrowdPrisma and other pipelines would be even greater on a corpus that is linguistically more complex and challenging (which is the domain where CrowdPrisma was developed, i.e. policy research). This recommendation is further supported by the fact that CrowdPrisma is the only verifiable method, that provides exact quotes to back up its topic assignments. It is also the only method that automatically constructs a two-level hierarchy of themes and topics.
  - If we have substantial domain knowledge of the text data we want to understand, and we can leverage LLMs (no data sensitivity issues), then a simple LLM based topic modelling pipeline can be a good choice, provided that we have the LLMOps capabilities to make it a robust.
  - If we have labelled data, then topic extraction is not needed obviously and a supervised approach might be best. Especially on linguistically simple text like this customer service dataset.
- Overall, the CrowdPrisma TextEngine offers a unique combination of features that can make it extremely desirable in certain topic modelling scenarios, especially in high-value knowledge work and research. From the currently available topic modelling pipelines, CrowdPrisma is the only viable option if:
  - the corpus is long and complex,
  - domain knowledge of the main themes and topics are not known a priori
  - a theme & topic hierarchy is desirable due to the complexity of the discussed ideas
  - multi-label assignment is needed (each piece of text can talk about more than one topic)
  - verifiable assignments are important (supporting quotes for checking the system’s output)