A Corpus and Evaluation for Predicting Semi-Structured Human Annotations

Andreas Marfurt[1,2,*], **Ashley Thornton**[3], **David Sylvan**[3], **Lonneke van der Plas**[1,4], and **James Henderson**[1]

[1]Idiap Research Institute, Switzerland
[2]EPFL, Switzerland
[3]Graduate Institute of International and Development Studies, Switzerland
[4]University of Malta, Malta

Abstract

A wide variety of tasks have been framed as text-to-text tasks to allow processing by sequence-to-sequence models. We propose a new task of generating a semi-structured interpretation of a source document. The interpretation is semi-structured in that it contains mandatory and optional fields with free-text information. This structure is surfaced by human annotations, which we standardize and convert to text format. We then propose an evaluation technique that is generally applicable to any such semi-structured annotation, called equivalence classes evaluation. The evaluation technique is efficient and scalable; it creates a large number of evaluation instances from a comparably cheap clustering of the free-text information by domain experts. For our task, we release a dataset about the monetary policy of the Federal Reserve. On this corpus, our evaluation shows larger differences between pretrained models than standard text generation metrics.

1 Introduction

General-purpose sequence-to-sequence models have achieved impressive results on conditional text generation (Radford et al., 2019; Brown et al., 2020), machine translation (Liu et al., 2020; Xue et al., 2021), and text summarization (Lewis et al., 2020; Zhang et al., 2020a). This has lead to their application to ever more tasks; as long as the task can be formalized in a text-to-text format, it can be processed by these models (Raffel et al., 2020).

We apply sequence-to-sequence models in a different setting: documents interpreting other documents. This phenomenon is pervasive in our daily lives, be it a critic reviewing a play or book, a website presenting highlights of a travel guide, or, as in this paper, a journalist writing an article about an organization's press release.

For social scientists, these reviews or articles present an interesting subject of study; they surface the author's interpretation of the original source document. With the tool of human annotations, domain experts can extract the core constituents and surface implicit information in these various interpretations to make them comparable.

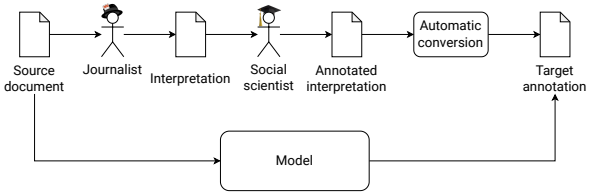

Figure 1: Our proposed interpretation task. A journalist creates an interpretation of a source document. A social scientist extracts and categorizes the relevant parts of the interpretation by means of annotation, which is converted into text-only format. The model learns the interpretation process by directly predicting the target annotation from the source document.

In this paper, we train models to learn this interpretation process (see Figure 1).[1] We describe how human annotations can be standardized and converted into a text-only format to serve as semi-structured targets in this interpretation prediction task. We introduce the *FOMC* dataset, a corpus about the monetary policy of the Federal Reserve, the central bank of the United States of America. The dataset contains source documents of greatly varying length, containing policy announcements such as press releases or speeches. The target interpretations are short, and consist of selected sentences taken from New York Times articles, which are then annotated by domain experts. We also devise a scalable evaluation technique for semi-structured outputs, which we call *equivalence classes evaluation*. Domain experts cluster highlighted text spans from the human annotations into equivalence classes, signifying their semantic inter-

*Correspondence to andreas.marfurt@idiap.ch.

[1]Our data, code and finetuned models are available at https://github.com/idiap/semi-structured-annotations.

Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pages 262 - 275

changeability. A generative model is then probed with a prefix and either a true continuation from the data or a wrong continuation from a different equivalence class. If the model learned the process of interpretation well, it will give higher probability to the true continuation. From a single clustering, we can automatically generate a large number of evaluation instances by sampling negative text spans. We train Transformer (Vaswani et al., 2017) sequence-to-sequence models with varying levels of pretraining on the FOMC dataset, and find that BART (Lewis et al., 2020) performs well on our equivalence classes evaluation and standard text generation evaluation metrics.

Our contributions are: 1) We introduce a new dataset on document interpretation, with semi-structured annotations and documents on the monetary policy of the Federal Reserve. 2) We introduce a method to convert these annotations into text format and apply generative text models. 3) We devise a scalable technique to evaluate models on the task of generating semi-structured outputs, by efficiently utilizing domain experts' grouping of text spans into equivalence classes. 4) We perform an evaluation with our technique, and showcase its flexibility with an in-depth error analysis.

2 Semi-Structured Human Annotations

We consider a setting where a long text is interpreted in a few sentences. The interpretation may select an aspect of the source text to focus on, and it may be opinionated. Each sentence is annotated by domain experts to surface and structure the important information.

2.1 Standardizing Human Annotations

We aim to standardize the human annotations into a general but flexible semi-structured format which should make it possible for NLP models to process them. In order to do so, we first have to define the possible annotation operations.

Our annotations are created from two operations: 1) marking spans with a label in order to categorize them, and 2) optionally commenting on a marked span to give context, paraphrase or make implicit information explicit.

2.2 Converting Annotations to Text

We convert annotations into a text-only format by inserting category-specific start and end tokens for each marked span. Overlapping or fully contained spans are allowed. We include the comments by adding them in parentheses (imitating a similar use in natural language) at the end of the respective marked span and before the category end token. An example annotation transformed to text format is shown in Figure 2.

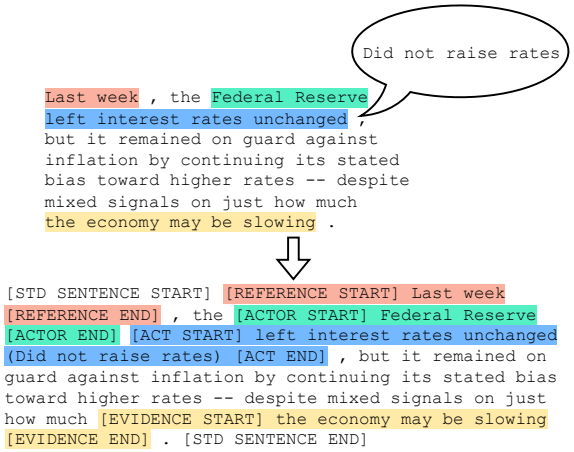

Figure 2: Example of the automatic conversion of an annotated interpretation into text format.

2.3 The Interpretation Task

We propose the task of generating the interpretations, including the human annotations, from the source documents. This task can be formalized as a sequence-to-sequence generation task, with pairs of a single source document x_i and one or more target annotations y_{ij} in text format, with $1 \leq j \leq m_i$. The multiple target annotations are equivalent to multiple references in traditional text generation tasks, i.e. they are all equally valid solutions to the task. In total, there are n source documents and $m = \sum_{i=1}^{n} m_i$ targets.

The targets y_{ij} contain marked spans of categories c from a predefined set of categories C. Some categories occur in every target, and some are optional, as illustrated in the paragraph *Annotation Categories* below.

2.4 The FOMC Dataset

We now present our dataset constructed according to the guidelines above. The source documents and targets were selected and annotated by domain experts. They are on the topic of the monetary policy of the Federal Open Market Committee (FOMC) of the Federal Reserve, the central bank of the United States of America, in the years from 1967 to 2018. The source documents are policy announcements

of the FOMC such as press releases, speeches, testimonies, Q&A sessions, or meeting minutes. The targets are sentences from news articles of the New York Times which conform to the requirements (see *Annotation Categories* below). An example is shown in Appendix D.

Data collection. Domain experts[2] searched the New York Times archives for articles on the monetary policy of the Federal Reserve. Candidate articles were searched for sentences that contain all mandatory categories described below. If a sentence is found, it is annotated by highlighting the categories and adding comments. All annotations are validated by a senior domain expert[3]. Sentences from the same article referencing the same source document are collected in a single target annotation. If multiple articles reference the same source document, one target annotation is created per article.

Annotation Categories. A selected sentence is called a *standardized sentence* in the corpus terminology. The mandatory and optional categories, as well as their purpose, are listed below:

- **Standardized sentence:** Mandatory. Marks the start and end of a target sentence.
- **Act:** Mandatory. Most often contains a comment. Marks an action (or non-action) on monetary policy. Example: "left interest rates unchanged (Did not raise rates)".
- **Actor:** Mandatory. Marks the entity performing the act. By design, this is exclusively the Federal Reserve or FOMC. Example: "Fed".
- **Reference:** Mandatory. Provides a link to the source document, which can be opaque in the article, e.g. saying that something happened yesterday. That source is systematically tracked down by the domain experts. Example: "yesterday's meeting".
- **Attribution:** Optional. Marks the individual advocating for the Federal Reserve to perform a certain action. Example: "Greenspan".
- **Motive:** Optional. Can appear multiple times. States the goal of an act. Example: "to fight inflation".

[2]PhD students in economics and political science at the Graduate Institute

[3]Ashley Thornton and David Sylvan

	Train	Valid	Test
Source documents	1342	167	169
Target annotations	3246	364	380
Mean targets/source	2.42	2.18	2.25
Max targets/source	36	17	16

Table 1: Number of examples in each split in the FOMC dataset.

Count	Source documents	Targets
(Std) Sentences	262.6 (± 688.6)	1.6
Words	6054.1 (± 12639.4)	123.6
Start/end tokens	-	18.2
Total tokens	6054.1 (± 12639.4)	141.8

Table 2: Mean length of source documents and target annotations in the FOMC dataset.

- **Evidence:** Optional. Can appear multiple times. States an observation, e.g. about the current economic state, that served as an incentive for the act. Example: "high oil prices".
- **Scope:** Optional. Marks the temporal scope of an act. Example: "by the end of the year".

Dataset statistics. We now show dataset statistics. First, the number of examples in each split (80%/10%/10% for train/validation/test) is presented in Table 1. Second, the number of tokens in source and target texts is shown in Table 2.

Filtering source documents. As is evident from Table 2, the source documents are generally very long. In contrast, the maximum number of input tokens that state-of-the-art models are pretrained on, lies between 512 in BERT (Devlin et al., 2019) and 1024 in BART (Lewis et al., 2020). This limitation is due to the quadratic complexity of self-attention in the Transformer architecture (Vaswani et al., 2017) and its resulting strain on computational resources.

As a consequence, there are two ways to define the prediction task for the FOMC dataset. The first one is to condition on the full text document, but devise models capable of handling very long inputs, such as adding a filtering module (used below) or using appropriate architectures such as the Longformer (Beltagy et al., 2020). The second option is to condition on a specific filtering of the source documents which reduces them to a length that can be processed by the chosen model. Alongside the data, we provide a script that allows for selecting sentences from the source document,

```
Evaluation: Act type
Category: Act
Equivalence class 1:
  - left interest rates
    unchanged (Did not
    raise rates)
Equivalence class 2:
  - decided to raise interest
    rates (Did raise rates)
  - voted to raise rates (Did
    raise rates)
Equivalence class 3:
  - lowered interest rates a
    quarter point (Cut rates)
```

Figure 3: Definition of an evaluation with its equivalence classes.

while satisfying the length restriction for a given tokenizer from the HuggingFace transformers library (Wolf et al., 2020). The selection logic can be set to either pick sentences from the top of the source document (*Lead* strategy), or to use an oracle that greedily picks sentences that maximize the length-normalized ROUGE-2 recall gain (*Oracle* strategy).

3 Equivalence Classes Evaluation

To evaluate a model on predicting the marked spans of individual annotation categories, we propose the *equivalence classes evaluation* as an efficient way of generating evaluation instances from domain experts' knowledge.

3.1 Definition

An evaluation selects a category c that it wants to evaluate, which in turn consists of 2 or more equivalence classes. The members of an equivalence class are marked spans of category c in the dataset. Members of the same equivalence class are semantically interchangeable in the target annotation, with respect to the objective of the evaluation. The members of all equivalence classes must be syntactically interchangeable, such that replacing one for the other still results in a grammatically correct sentence. An example is given in Figure 3.

3.2 Creating Evaluation Instances

Evaluation instances are then created by searching target annotations in the validation/test set for a member of an equivalence class. If one is found, an evaluation instance is created consisting of 1) the prefix y_{prefix} up until the selected span, 2) the selected span $a^{(\text{pos})}$ as the true (positive) continuation, and 3) a randomly selected span $a^{(\text{neg})}$ from a different equivalence class as the false (negative) continuation. $a^{(\text{neg})}$ is chosen by uniformly sampling a negative equivalence class, and then uniformly sampling one of its members. An example is shown in Figure 4, where $a^{(\text{pos})}$ is in equivalence class 1 of the example evaluation in Figure 3, and $a^{(\text{neg})}$ has been sampled as the first member of equivalence class 2. Any other member of equivalence classes 2 or 3 could have been chosen as well.

y_{prefix}	[REFERENCE START] Last week [REFERENCE END] , the [ACTOR START] Federal Reserve [ACTOR END] [ACT START]
$a^{(\text{pos})}$	left interest rates unchanged (Did not raise rates)
$a^{(\text{neg})}$	decided to raise interest rates (Did raise rates)

Figure 4: Equivalence classes evaluation instance with prefix y_{prefix}, a positive continuation $a^{(\text{pos})}$ and a negative continuation $a^{(\text{neg})}$.

For a single match of a positive span in the evaluation set, one can create a large number of evaluation instances by sampling negative continuations without replacement.

Optionally, the positive span $a^{(\text{pos})}$ can be replaced by a different member of the same equivalence class (for equivalence classes with more than one member). This can help mitigating lexical inaccuracies that can arise from replacing a span with another, which otherwise only exist for $a^{(\text{neg})}$.

Relation to Pattern-Exploiting Training. In Schick and Schütze (2021a), Pattern-Exploiting Training (PET) is introduced. The concept of verbalizers is similar to equivalence classes. In their work, verbalizers are manually predefined single tokens that represent a class label.[4] Our equivalence classes consist of expert-selected multi-word spans from the data, that each represent the concept of their equivalence class. Equivalence classes are multi-faceted: They determine both a semantic concept and a grammatical structure, and are always defined with respect to a certain aspect under evaluation.

3.3 Model Evaluation

To evaluate the generative model, we obtain the probability p_θ it assigns to $a^{(\text{pos})}$ and $a^{(\text{neg})}$ by getting its next-token probabilities given the prefix y_{prefix}. We apply teacher-forcing and obtain the

[4] In their follow-up work, they extend verbalizers to multiple tokens (Schick and Schütze, 2021b).

probabilities autoregressively, extending the prefix with the previous token at each turn. The probability of the entire span is computed as

$$p_\theta(a) = \prod_{i=1}^{l} p_\theta(a_i | y_{\text{prefix}}, a_{<i}) \quad (1)$$

where $a \in \{a^{(\text{pos})}, a^{(\text{neg})}\}$, and l is the length of a. The model solves an instance correctly if $p_\theta(a^{(\text{pos})}) > p_\theta(a^{(\text{neg})})$.

If the lengths of $a^{(\text{pos})}$ and $a^{(\text{neg})}$ are substantially different, the value of p_θ could be determined more by the difference in length than in semantics. We avoid this during sampling of $a^{(\text{neg})}$ by restricting the maximum difference in number of words between $a^{(\text{neg})}$ and $a^{(\text{pos})}$ to 2.

3.4 In-Depth Analysis

The equivalence classes evaluation also allows for an in-depth error analysis. First, we can test specific properties for a category, such as how well the model handles negation in acts. Second, we can break down an evaluation's score by combinations of equivalence classes, and identify the hardest combinations for the model. We show examples of such analyses in Section 5.3.

Data augmentation. As an added benefit, equivalence classes give rise to a simple training data augmentation method. We create additional training examples from equivalence classes by exchanging the ground-truth highlighted span with a different one from the same equivalence class. We postpone testing the efficacy of this data augmentation method to future work.

4 Experiments

In our experiments, we use the FOMC dataset described in Section 2.4.

4.1 Equivalence Classes

The equivalence classes for our evaluations were proposed by one of the authors[5] and validated by the same senior domain experts as for the FOMC dataset described in Section 2.4. We create an evaluation for each of the following 5 categories: act, attribution, motive, evidence, and scope. We add one evaluation for the act comments, without the act itself. Furthermore, we create additional in-depth evaluation examples for modal verbs and negation, which our domain experts are especially interested in. We create separate evaluations for modal verbs in positive (e.g. *should*) and in negative formulation (e.g. *might not*), to avoid confounding with the effect of negation. These 3 evaluations (positive modal verbs, negative modal verbs, negation) are created for acts without comments, act comments, and acts concatenated with comments.

One evaluation instance is created for each example in the evaluation set that contains any member of the evaluation's equivalence classes. If the evaluation instances n have not reached 100 yet, $\lfloor \frac{100}{n} \rfloor$ negative spans $a^{(\text{neg})}$ are sampled per instance, such that the total number is close to 100. If more than 100 matches are found, all of them are included in the evaluation with one randomly sampled negative span. We do not replace positive spans. This procedure generates 1974 total evaluation instances from the validation set, and 2104 from the test set. The general evaluations of the 5 categories plus the act comments (excluding in-depth evaluations) contain 818 evaluation instances from the validation set, and 886 from the test set. The smallest category (motive) has 74 and the largest (act labels) has 336 test evaluation instances.

4.2 Standard Text Generation Metrics

Since our task is a sequence-to-sequence task, we also report standard text generation metrics. If not mentioned otherwise, we compute the following metrics for generated annotations without special tokens (category start and end tokens).

ROUGE. ROUGE (Lin, 2004) is a textual overlap metric which is widely used in text summarization, a task with strong connections to ours. As is common in summarization, we report ROUGE-1/2/L as the unigram and bigram overlap, and the longest common subsequence, respectively. We compute ROUGE with and without special tokens, as we want to see both how well the model generates the annotations as well as the original article.

BERTScore. We use BERTScore (Zhang et al., 2020b) as a semantic similarity metric between the generated and reference target annotations. We do not use idf-importance weighting, and we use baseline rescaling.[6] If multiple target annotations are present, the maximum similarity is reported, as proposed by the authors.

[5] Andreas Marfurt

[6] Evaluation hash: `roberta-large_L17_no-idf_version=0.3.11(hug_trans=4.6.1)-rescaled`

Distinct bigrams. We report the distinct bigrams in the generated target annotations. This metric checks if the model produces overly generic and repetitive outputs. A higher number of distinct bigrams corresponds to higher lexical diversity in the output and is desirable.

Novel bigrams. Novel bigrams measure the percent of bigrams in a generated annotation that do not appear in the filtered source document that serves as input text. This metric measures the extractiveness of the model, i.e. its tendency to copy text from the input.

4.2.1 Annotation Category Metrics

We add annotation category-specific metrics to the text generation metrics. These metrics are designed to detect if any category or the target format are ignored by the model.

Category counts. We report the mean and standard deviation of each annotation category's occurrence over the generated target annotations.

Categories correctly closed. This evaluation measures the percent of annotation spans that are correctly encompassed by a category start and end token. This evaluation shows whether the decoder correctly learned to generate in the target format.

4.3 Filtering Source Documents

As detailed in Section 2.4, the source documents are much longer than current Transformer models with quadratic self-attention complexity can process. However, we conjecture that only very specific parts of these documents are needed to generate the comparably very short target annotations (see Table 2). On top of mentioned filtering strategies, we train a filtering model. For this purpose, we finetune a BERT model (Devlin et al., 2019) for sequence classification.[7] We split long inputs at sentence boundaries into chunks of at most 512 tokens, and then predict whether to keep the sentences in the current chunk.

We train the model with a cross-entropy loss between the predictions and the oracle selection described in Section 2.4. We train with a batch size of 5 for 10 epochs, but stop early when the F1 score on the validation set no longer improves. We use the same learning rate schedule as for the generative models described below, with a maximum learning rate of 1e-3. During inference, we select the sentences with the highest logits until we reach the token limit. The selected sentences are concatenated in the order in which they appear in the source document. We name this filtering model *FilterBERT*.

[7] We use the standard implementation in the HuggingFace transformers library (Wolf et al., 2020).

4.4 Generative Models

For our generative models, we rely on the Transformer architecture (Vaswani et al., 2017), and compare finetuning differently pretrained models.

Transformer. We use a randomly initialized Transformer encoder-decoder to test the effect of skipping pretraining. Our implementation of the Transformer is the same as the BERT model below.

BERT. We finetune a pretrained BERT encoder (Devlin et al., 2019) and train a randomly initialized Transformer decoder, as proposed in Liu and Lapata (2019). Unless otherwise mentioned, we use the base model size.

BART. We finetune the BART model (Lewis et al., 2020) as a proponent of a jointly pretrained encoder and decoder.

Training details. Training steps and learning rate hyperparameters were selected on the validation set with a grid search with exponential step sizes. We train our models for a maximum of 10 (Transformer/BERT) or 20 (BART) epochs, which corresponds to 8000 or 16000 steps with a batch size of 4, respectively. We stop training early if the validation loss does not improve any further. We set the maximum learning rate to 1e-4 for randomly initialized parameters, and 1e-5 for pretrained ones. Exceptionally for BART, we use a learning rate of 1e-6 for the tied input/output embeddings. We warm up the learning rate for a tenth of the total epochs, with a linear increase from 1/100-th of the maximum learning rate, and then a linear decay back down to the starting point. We use the Adam optimizer (Kingma and Ba, 2015).

Generation details. For our evaluation of text generation metrics (see Section 4.2), we generate text with beam search. We use 5 beams, a minimum generation length of 50 tokens and a maximum of 500, no length penalty, and no n-gram blocking (Paulus et al., 2018).

Model	Act	Act comments	Attribution	Motive	Evidence	Scope	Mean
Transformer	93.18%	93.45%	94.79%	66.22%	43.68%	50.00%	73.55%
BERT	97.73%	94.64%	97.16%	66.22%	45.98%	54.44%	76.03%
BART	98.86%	96.13%	97.16%	71.62%	81.61%	80.00%	87.56%

Table 3: Accuracy of main equivalence classes evaluations.

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	Distinct bigrams	Novel bigrams
References	-	-	-	-	9961	82.68%
Transformer	31.89	10.27	25.06	0.72	214	87.61%
BERT	41.09	18.31	31.63	19.00	965	82.75%
BART	42.73	20.64	33.08	26.78	3011	73.62%

Table 4: Text generation evaluation results. ROUGE is computed on targets including special tokens.

5 Results

5.1 Equivalence Classes Evaluation

Our main results for the general equivalence classes evaluations on the 5 categories plus act comments are shown in Table 3. The BART model with a jointly pretrained encoder and decoder substantially outperforms the Transformer and BERT models. The act, act comments and attribution evaluations are solved nearly perfectly, but the others are harder. For the evidence evaluation, Transformer and BERT perform substantially below the random baseline, which would achieve 50% in expectation. We analyze the case of the evidence evaluation further in Section 5.3.

5.2 Text Generation Metrics

We show the results of text generation metrics in Table 4. Again, BART outperforms the other models. The low scores in BERTScore and distinct bigrams (excluding special tokens) indicate that the Transformer fails to generate diverse and topical target annotation sentences. However, the comparably high ROUGE scores (including special tokens) show that it learns to generate the target format well, which is also supported by the last column of Table 5. BART generates the most diverse and topical target annotations, and is also the most extractive method, showing that it makes use of the input document.

In Table 5 in the appendix, we show the mean and standard deviation of each category's annotation counts for our three models. BERT produces outputs that stay closest to the number of category annotations of the reference target annotations. BART under-generates all categories, which can be partially explained by it not having learned to open and close category spans reliably. The combination of not having seen the format during pretraining and a lower decoder learning rate, which was helpful for the other tasks, explains why BART performs worse than the models with randomly initialized Transformer decoders.

5.3 In-Depth Analysis

Table 6 in the appendix shows a selection of equivalence classes evaluations where equivalence classes were built for a specific purpose. In our evaluations, these measure performance on act modal verbs (e.g. *raised rates* vs. *might raise rates*) and act negation, both aspects that are of high importance to our domain experts. We can see that negation is handled well by all models, and that modals are substantially harder for acts, but not for act comments (where the act is part of the prefix). Acts with comments (last column) do not necessarily make the task easier than acts without comments (second column).

We also perform a qualitative in-depth analysis of the evidence evaluation for the BART model. To that effect, we count the percentage of evaluation instances the model gets wrong for each pair of equivalence classes, which is shown in the confusion matrix in Figure 5. The number in each square corresponds to the number of mistakes in the evaluation. Some of the mistakes occur for the following pairs of equivalence classes, where $a^{(\text{pos})}$ is taken from the first, and $a^{(\text{neg})}$ from the second:

- deflation – low/declining inflation
- cooling housing market – tightening credit market (full example in Appendix E)
- high unemployment – high oil prices
- high unemployment – weak economic activity

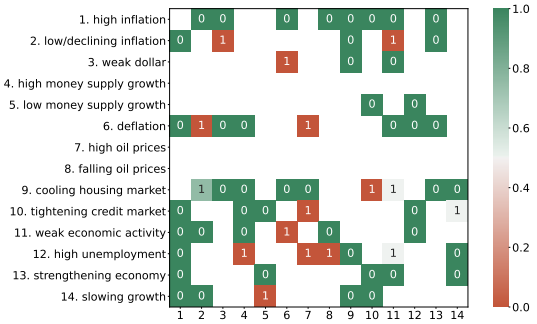

Figure 5: Confusion matrix of BART's accuracy on pairs of evidence equivalence classes. The equivalence class for $a^{(\text{pos})}$ is on the y-axis, the one for $a^{(\text{neg})}$ on the x-axis. Each cell contains the number of mistakes the model makes for that combination. Empty cells do not have a corresponding pair in the evaluation.

- slowing growth – low money supply growth

The mentioned combinations of economic processes are correlated or even co-occurring, making it difficult for the model to distinguish the positive from the negative span. In these cases of close semantic similarity, the model may fall back to ranking candidate text spans higher based on e.g. their frequency in the training data, where inflation is one of the dominant subjects. For other combinations, such as weak dollar – deflation, the model just makes mistakes.

5.4 Ablation Study

We perform ablation studies with respect to model size, filtering strategy and source document input length. The tables with the results have been moved to Appendix C.

Model sizes. In the results shown so far, BART has outperformed the Transformer and BERT. However, those models operate with 247 million parameters (size of BERT-base), while BART has 406 million. In Table 7, we see that increasing BERT's parameters to the size of BERT-large only provides small to no benefits. BART still outperforms BERT-large, even with almost half of the parameters, due to – as we believe – the beneficial initialization from joint encoder-decoder pretraining. This is especially valuable on the FOMC dataset, which has comparably few training instances.

Filtering strategies. In Section 4.3, we introduced the FilterBERT model for identifying and selecting salient sentences from long source documents. As stated in Section 2.4, together with the dataset, we make available a script for filtering source documents with either the Lead or the Oracle strategy. The former selects sentences from the top of the source document, the latter selects those that most increase the length-normalized ROUGE-2 recall with the target annotations. In Table 8, we see that Oracle filtering generally performs best on generation metrics, but not on equivalence classes. The FilterBERT model outperforms the Lead strategy for BART but not for generation metrics on BERT. In general, the differences between the generative models are much larger than between the filtering strategies.

Source document input length. Finally, since BART has the ability to process inputs of up to 1024 tokens in length, we evaluate how that compares to the input length of 512 tokens that we have used so far. The results in Table 9 show that for the Lead filtering strategy, longer inputs benefit all metrics except ROUGE-2. With Oracle filtering, ROUGE-2 and distinct bigram evaluations perform slightly worse with longer inputs, while the rest improve. In summary, the additional input sentences only make a small difference for BART.

6 Related Work

To the best of our knowledge, our setting, task, and evaluation have not been studied in prior work.

Evaluation. The closest approach to our equivalence classes evaluation is the concept of verbalizers in Pattern-Exploiting Training (PET) (Schick and Schütze, 2021a,b), the relation to which we already discussed in Section 3.2. The biggest difference to our approach is that PET's verbalizers are limited to a small, bounded set of predefined single tokens or few-token spans, while our equivalence classes are unbounded, and their members are collected from the data without restrictions on length or content.

Other work has also tried to make human annotations more efficient, e.g. for importance judgements of sentences in multi-domain summarization (Jha et al., 2020), or multi-task information extraction (Bikaun et al., 2022). AnnIE builds fact synsets to speed up open information extraction (Friedrich et al., 2022).

Aspect-oriented summarization. Interpretations may focus on certain aspects in the source documents, making them somewhat similar to aspect-oriented summarization. AspectNews

(Ahuja et al., 2022) and SPACE (Angelidis et al., 2021) are two recent datasets with accompanying models.

News summarization. Since our interpretations are excerpts of New York Times articles, news summarization is relevant to our work as well. This is a very active field of research, with multiple large-scale datasets (e.g. CNN/DM (Hermann et al., 2015; Nallapati et al., 2016), XSum (Narayan et al., 2018), among others). A lot of methods have been tried on these datasets. BART (Lewis et al., 2020) and PEGASUS (Zhang et al., 2020a) have shown some of the best results for fine-tuned models.

7 Conclusion

We have devised a method to convert semi-structured human annotations into text format. We then introduced a task of predicting annotated interpretations of source documents that can be tackled with sequence-to-sequence models. We presented a human-annotated corpus about the monetary policy of the Federal Reserve. Our equivalence classes evaluation is an efficient technique to create a large number of targeted evaluation instances from a comparably cheap clustering by domain experts. We use this technique to evaluate state-of-the-art generative models on our task, and find that it shows larger differences between pretrained models than standard text generation metrics. In further in-depth analyses, the equivalence classes evaluation tests the models for specific properties, such as how they handle negation, and detects why models struggle to correctly rank alternative text spans of certain human annotation categories.

Limitations

In the following we discuss some limitations of our paper.

Subjectivity of the annotation process. Even though annotation protocols can be standardized and outputs aggregated over multiple annotators, the process of annotation remains subjective. In our interpretation task, social scientists extract and categorize information, providing additional context where necessary, and all annotations are validated by a senior domain expert. The models trained on the data will focus on the aspects that the annotators deemed important. This is not inherently a bad thing. Human annotation is a flexible tool that a different set of annotators could use to highlight other aspects of the data. Note that this is a separate consideration from reproducibility of our results, which we enable by open-sourcing our data, code and models.

Application to other domains. We have yet to establish transferability of our allowed set of annotations and task setup to other domains. While we expect our procedure to be general enough to work in different areas, this paper only uses a single corpus about macroeconomics. The reason for the limitation to one corpus is the high cost of finding relevant interpretation documents, performing the extraction and annotation, and standardizing the resulting annotations.

Equivalence classes creation. While the creation of equivalence classes is less expensive than directly creating evaluation examples, it still requires manual effort by domain experts, which is an expensive resource. This could be alleviated with an automatic method to obtain equivalence classes. In theory, the identification of candidate members of equivalence classes should be facilitated by the category annotations. The two member properties of 1) semantic interchangeability within equivalence classes and 2) syntactic interchangeability across equivalence classes could potentially be judged by a strong language model.

Syntactic structure of equivalence class members. Syntactic interchangeability is a requirement on equivalence class members within one equivalence classes evaluation. This limits us to one syntactic construction per evaluation. We select the most common one in each category to obtain a large enough number of evaluation instances. As a consequence, the model will not be tested on different syntactic structures. Unfortunately, testing all possible syntactic constructions suffers from 1) a data sparsity problem, where not enough examples of the same construction occur in the data, and 2) a large increase in manual effort required to construct one evaluation per syntactic structure.

Ethical Considerations

Since this work uses pretrained language models, it inherits the problems of those models with respect to reproducing biased or offensive content present in the pretraining data. We finetune our models on the FOMC dataset, which consists of policy announcements of the FOMC and news article sentences of the New York Times. Both of these

sources can be considered trustworthy and careful with respect to the language that they use, in contrast to general text on the web that was present in BART's pretraining data. The topic of our dataset is the monetary policy of the Fed. Non-topical content was filtered in the data collection stage. All included news article sentences (which form the targets of our finetuning) were carefully selected and annotated. We therefore expect not to have introduced additional ethical issues with our dataset or finetuning. It should be noted that the dataset dates from 1967 to 2018, so it spans different historical contexts.

The annotations were performed by researchers of the Graduate Institute in their capacity as PhD students, postdocs and professors.

Acknowledgments

This work was supported as a part of the grant Automated interpretation of political and economic policy documents: Machine learning using semantic and syntactic information, funded by the Swiss National Science Foundation (grant number CRSII5_180320), and led by the co-PIs James Henderson, Jean-Louis Arcand and David Sylvan. We would also like to thank Maria Kamran, Alessandra Romani, Julia Greene, Clarisse Labbé, Shekhar Hari Kumar, Claire Ransom, Daniele Rinaldo, Eugenia Zena and Raphael Leduc for their invaluable data collection and annotation efforts.

References

Ojas Ahuja, Jiacheng Xu, Akshay Gupta, Kevin Horecka, and Greg Durrett. 2022. ASPECTNEWS: Aspect-oriented summarization of news documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6494–6506, Dublin, Ireland. Association for Computational Linguistics.

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. Extractive opinion summarization in quantized transformer spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Tyler Bikaun, Michael Stewart, and Wei Liu. 2022. QuickGraph: A rapid annotation tool for knowledge graph extraction from technical text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186.

Niklas Friedrich, Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Mathias Niepert, and Goran Glavaš. 2022. AnnIE: An annotation platform for constructing complete open information extraction benchmark. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 44–60, Dublin, Ireland. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Rahul Jha, Keping Bi, Yang Li, Mahdi Pakdaman, Asli Celikyilmaz, Ivan Zhiboedov, and Kieran McDonald. 2020. Artemis: A novel annotation methodology for indicative single document summarization. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 69–78, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *https://openai.com/blog/better-language-models*.

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*, 21:1–67.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammed Saleh, and Peter J. Liu. 2020a. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. BERTScore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Category Counts

The mean and standard deviation of category counts, for the references as well as the model generations, are listed in Table 5.

B In-Depth Results

Selected results of the in-depth analysis are shown in Table 6.

C Ablation Results

We present the results of our ablation study on model sizes in Table 7, on filtering strategies in Table 8, and on source document input lengths in Table 9.

D Full FOMC Example

We present the first example from the FOMC test set. We show the source document filtered with FilterBERT, the target annotation and BART's prediction. At the end, we show the generation scores for this example.

Filtered source document: FEDERAL RESERVE press release For Use at 4:30 p.m. August 22, 1986 The Federal Reserve Board and the Federal Open Market Committee today released the attached record of policy actions taken by the Federal Open Market Committee at its meeting

on July 8-9, 1986. Such records for each meeting of the Committee are made available a few days after the next regularly scheduled meeting and are published in the Federal Reserve Bulletin and the Board's Annual Report. The summary descriptions of economic and financial conditions they contain are based solely on the information that was available to the Committee at the time of the meeting. Attachment RECORD OF POLICY ACTIONS OF THE FEDERAL OPEN MARKET COMMITTEE Meeting Held on July 8-9, 1986 Domestic policy directive The information reviewed at this meeting indicates that economic activity has expanded at a relatively slow pace recently. The intermeeting range for the federal funds rate was reduced to 5 to 9 percent. Other interest rates rose early in the period but then retreated amid signs of weakness in the economies of the United States and some of its major trading partners, renewing expectations of a discount rate cut in the near future. Since the May meeting short-term market rates had declined 10 to 40 basis points on balance. In their discussion of policy implementation for the weeks immediately ahead, Committee members took account of the likelihood that the discount rate would be reduced within a few days after the meeting. Against the background of sluggish expansion in economic activity and a subdued rate of inflation, most of the members believed that some easing was desirable and they indicated a preference for implementing the easing, at least initially, through a lower discount rate rather than through open market operations. In one view, a cut in the discount rate might need to be accompanied by some increase in the degree of pressure on reserve positions, pending evaluation of further economic and financial developments. The reduction was viewed as a technical adjustment that would provide a more symmetrical range around a lower federal funds rate that could be expected to emerge following the anticipated reduction in the discount 7/8-9/86 - 18 rate. Most short-term interest rates have declined on balance since the May 20 meeting of the Committee. In the implementation of policy for the immediate future, the Committee seeks to decrease somewhat the existing degree of pressure on reserve positions, taking account of the possibility of a change in the discount rate.

Target annotation: [STD SENTENCE START] Policymakers at the [ACTOR START] Federal Reserve [ACTOR END] [ACT START] decided at their July meeting to loosen credit conditions (Loosened monetary policy) [ACT END] [MOTIVE START] in an effort to stimulate the sluggish economy [MOTIVE END], according to [REFERENCE START] minutes [REFERENCE END] of the meeting released today. [STD SENTENCE END] [STD SENTENCE START] Members of the [ACTOR START] Federal Open Market Committee (Fed / FOMC) [ACTOR END] [REFERENCE START] voted [REFERENCE END] 10 to 1 to follow a strategy that would push interest rates lower, [ACT START] despite [ATTRIBUTION START] objections from one member (Should not loosen monetary policy) [ACT END] (One member of the FOMC) [ATTRIBUTION END] that [EVIDENCE START] such a course might threaten renewed inflation later [EVIDENCE END]. [STD SENTENCE END] [STD SENTENCE START] Thomas C. [ATTRIBUTION START] Melzer [ATTRIBUTION END], president of the St. Louis [ACTOR START] Federal Reserve Bank [ACTOR END], [ACT START] cast the single dissenting vote (Should not loosen money supply) [ACT END]. The minutes said Mr. Melzer [REFERENCE START] expressed concern [REFERENCE END] that [EVIDENCE START] looser Fed controls could initiate renewed inflation [EVIDENCE END] and [EVIDENCE START] weaken the dollar on foreign exchange markets [EVIDENCE END]. [STD SENTENCE END]

BART prediction: [STD SENTENCE START] The [ACTOR START] Federal Reserve's Open Market Committee (Fed) [ACTOR END] [ACT START] voted unanimously at its July 8-9 meeting to ease monetary policy further (Might cut rates, in future) [ACT END], according to [REFERENCE START] minutes [REFERENCE END] of the session released today. [STD SENTENCE END]

ROUGE-1/2/L (including category markers): 31.68/17.00/28.71
ROUGE-1/2/L (excluding category markers): 30.38/15.38/27.85
BERTScore: 27.29
Novel bigrams: 84.38%
Closed correctly: 100.00%

E Equivalence Classes Evaluation Example

We present an evaluation instance from the equivalence classes evaluation for the evidence category. BART got this example wrong, i.e. judged the negative continuation $a^{(\text{neg})}$ as more likely than the positive $a^{(\text{pos})}$ (appears in Figure 5, positive class 9, negative class 10).

$y_{\textbf{prefix}}$**:** [STD SENTENCE START] Ben S. [ATTRIBUTION START] Bernanke [ATTRIBUTION END] , the chairman of the [ACTOR START] Federal Reserve [ACTOR END] Board, [REFERENCE START] declared [REFERENCE END] on Friday that the central bank [ACT START] "stands ready to take additional actions as needed" (Might cut rates, in future) [ACT END] [MOTIVE START] to prevent the chaos in mortgage markets from derailing the broader economy [MOTIVE END] . Mr. Bernanke avoided any specific promise to lower the central bank's benchmark federal funds rate at its next policy meeting on Sept. 18. But he acknowledged [EVIDENCE START]
$a^{(\textbf{pos})}$**:** the dangers posed by the twin storms in housing and mortgage lending
$a^{(\textbf{neg})}$**:** credit was becoming harder to get for both consumers and businesses

Model	Std sent	Act	Actor	Reference	Attribution	Motive	Evidence	Scope	Closed correctly
References	1.60 (± 0.93)	1.60 (± 0.93)	1.60 (± 0.93)	1.60 (± 0.93)	0.87 (± 1.18)	0.39 (± 0.72)	1.22 (± 1.41)	0.21 (± 0.45)	100.00%
Transformer	2.69 (± 0.62)	2.69 (± 0.62)	2.69 (± 0.62)	2.60 (± 0.64)	0.02 (± 0.22)	0.08 (± 0.31)	0.05 (± 0.22)	0.01 (± 0.11)	100.00%
BERT	1.63 (± 0.75)	1.63 (± 0.75)	1.63 (± 0.75)	1.63 (± 0.77)	0.73 (± 0.73)	0.43 (± 0.62)	0.07 (± 0.26)	0.14 (± 0.38)	99.97%
BART	1.55 (± 0.70)	0.79 (± 0.71)	1.22 (± 0.84)	1.37 (± 0.68)	0.29 (± 0.55)	0.02 (± 0.17)	0.13 (± 0.44)	0.05 (± 0.21)	69.44%

Table 5: Mean and standard deviation of category counts.

Model	Act negation	Act modals (pos)	Act comment modals (pos)	Act with comment modals (pos)
Transformer	89.58%	70.65%	93.81%	68.13%
BERT	93.75%	70.65%	93.81%	69.23%
BART	95.83%	89.13%	97.94%	80.22%

Table 6: Accuracy on selected in-depth equivalence classes evaluations.

Model	Parameters	EQ mean	ROUGE			BERTScore	Distinct bigrams
			R-1	R-2	R-L		
Transformer	247M	73.55%	31.89	10.27	25.06	0.72	214
BERT-base	247M	76.03%	41.09	18.31	31.63	19.00	965
BERT-large	771M	75.76%	41.26	17.91	31.39	19.30	1232
BART	406M	87.56%	42.73	20.64	33.08	26.78	3011

Table 7: Selected evaluation metrics for different model sizes.

Model	Filter model	EQ mean	ROUGE			BERTScore	Distinct bigrams
			R-1	R-2	R-L		
BERT	FilterBERT	76.03%	41.09	18.31	31.63	19.00	965
BERT	Lead	75.51%	41.27	18.74	31.44	19.59	1012
BERT	Oracle	75.15%	41.38	18.54	32.05	19.96	1010
BART	FilterBERT	87.56%	42.73	20.64	33.08	26.78	3011
BART	Lead	86.90%	41.56	19.79	32.09	25.15	1976
BART	Oracle	87.16%	44.04	21.84	33.87	26.98	3528

Table 8: Selected evaluation metrics for different filtering strategies.

Model	Filter model	Input tokens	EQ mean	ROUGE			BERTScore	Distinct bigrams
				R-1	R-2	R-L		
BART	Lead	512	86.90%	41.56	19.79	32.09	25.15	1976
BART	Lead	1024	87.12%	42.39	19.64	32.20	25.44	2421
BART	Oracle	512	87.16%	44.04	21.84	33.87	26.98	3528
BART	Oracle	1024	89.15%	44.81	21.36	34.77	27.79	3296

Table 9: Selected evaluation metrics for different source document input lengths.