# Automatic Fact-Checking with Frame-Semantics

Anonymous ACL submission

## Abstract

We propose a novel paradigm for automatic fact-checking that leverages frame semantics to enhance the structured understanding of claims, addressing the challenges posed by misinformation in today's information ecosystem. To support this approach, we introduce a pilot dataset of real-world claims extracted from PolitiFact, specifically annotated for large-scale structured data. This dataset underpins two case studies: the first investigates voting-related claims using the Vote semantic frame, while the second explores various semantic frames and data sources from the Organisation for Economic Co-operation and Development (OECD). Our findings demonstrate the effectiveness of frame semantics in improving evidence retrieval, indicating a meaningful advancement in automatic fact-checking capabilities. Finally, we conducted a survey of frames evoked in fact-checked claims, identifying high-impact frames to guide future research.

## 1 Introduction

The proliferation of misinformation presents a significant challenge to the modern information ecosystem. Mitigating the spread of fake news has become a critical concern for researchers, policymakers, and technology developers alike. In response, automatic fact-checking has emerged as an important research area, with the goal of reducing the burden on human fact-checkers by automating key steps in the fact-checking pipeline.

Many previous studies on automatic fact-checking have utilized unstructured data, such as fact-checks from trustworthy sources (Wang, 2017), to perform claim matching (Shaar et al., 2020) and use large language models to provide verdicts and explanations (Cheung and Lam, 2023b; Singhal et al., 2024a; Khaliq et al., 2024). Some works have also studied structured data, such as tabular data from Wikipedia (Chen et al., 2020; Aly et al., 2021) or scientific documents (Wang et al., 2021; Akhtar et al., 2022). These works have been studied on both simple claims extracted from Wikipedia (Bouziane et al., 2021) and real-world claims (Wang et al., 2021; Akhtar et al., 2022).

Previous approaches (Ye et al., 2023; Karagiannis et al., 2020; Jo et al., 2019) have found success with using SQL to query Wikipedia tables, which have an average of 13 rows per table (Chen et al., 2020). Due to a lack of existing datasets, no previous work has studied automatic fact-checking on real-world claims using large-scale structured data. To overcome the limitations of previous methods, we introduce a novel pilot dataset of real-world claims extracted from PolitiFact [1] annotated for large-scale structured data.

Our proposed pilot dataset serves as the foundation for two case studies focused on leveraging frame semantics for automatic fact-checking. Frame semantics (Ruppenhofer et al., 2016; Baker et al., 1998) is a linguistic framework that explores how language encodes meaning through structured representations called frames. These frames capture the essential elements and relationships in various situations, enhancing the understanding of context and intent behind statements.

In the first case study, we utilize the Vote semantic frame (Arslan et al., 2020), which represents how an *Agent* (e.g., a politician) interacts with a particular *Issue* (e.g., a proposed bill) through voting. Each frame includes specific components known as frame elements, which provide additional context about the roles and relationships involved, such as the voting member, the bill at stake, and the outcome of the vote. This case study includes 79 claims along with their corresponding congressional members and bills, with an average of 4,230 votes per congress member extracted from official U.S. voting records.

[1] https://www.politifact.com/

The second case study broadens the scope by exploring a variety of additional semantic frames and a diverse collection of datasets from the Organisation for Economic Co-operation and Development (OECD), which contain over 400 tables with an average of 596,000 rows and 13 columns. We annotated 68 OECD-related claims along with their corresponding frames and mapped each claim to the OECD statistics used to fact-check them. These case studies help to identify the challenges associated with automated fact-checking using large-scale structured data.

Our findings demonstrate the value of frame semantics in automatic fact-checking by enabling a structured and explainable understanding of claims. For instance, on voting claims, we found that using frame elements extracted from claims, instead of the entire claim itself, improved evidence retrieval, leading to a +2.1-point increase in recall@10 for voting claims. Similarly, on OECD claims, we saw a +7.3-point increase in recall@5 when frame elements were used to identify relevant OECD tables. These improvements illustrate how frame-semantics can enhance retrieval by guiding systems toward relevant structured data.

To understand which frames are most commonly evoked in factual claims, we surveyed claims fact-checked by PolitiFact and found a heavy skew towards a few specific frames, including Taking_sides, Speech, Change_position_on_a_scale. These insights guided the selection of frames for our case studies and enable future works to study high-impact frames.

To summarize, our contributions are as follows:

- We proposed a novel paradigm for automatic fact-checking using frame-semantics.
- We developed a pilot dataset for automatic fact-checking of real-world claims using large-scale structured data.[2]
- We conducted a novel survey of frames evoked in PolitiFact fact-checks, allowing researchers to target high-impact frames for future studies.
- We conducted two case studies on the efficacy of frame-semantics in automatic fact-checking and released a public demo.[3]

## 2 Related Works and Background

Recent advances in automatic fact-checking have been largely driven by the integration of large language models (LLMs) and retrieval-augmented generation (RAG) pipelines. Wang et al. (2024) proposed a unified framework for LLM-based systems which utilizes an internal mechanism to determine which of a selection of LLM-base models should be used to fact-check a particular claim. RAGAR (Khaliq et al., 2024) enhances fact-checking by using multimodal inputs and iterative reasoning. Similarly, FactLLaMA (Cheung and Lam, 2023a) combines pre-trained LLaMA models with external evidence retrieval to verify claims. LLM-Augmenter (Peng et al., 2023) integrates external knowledge sources and providing feedback to improve model accuracy. Singhal et al. (2024b) mitigates misinformation generated by RAG pipelines via re-ranking retrieved documents according to a credibility score.

For structured data, many approaches have been studied using fine-tuning methods (Zhao et al., 2022; Gu et al., 2022; Jiang et al., 2022) and LLM-based methods (Ye et al., 2023; Chen et al., 2021; Cheng et al., 2022) for fact verification. Dater (Ye et al., 2023) is a detailed fact verification system which decomposes claims into sub-questions using LLMs to simplify the fact-checking process and is the best-performing model on the TabFact (Chen et al., 2020) dataset. In this work, we do not focus on fact verification; however, we choose to utilize LLM-based methods for our fact verification step as they do not require retraining the model when new data is used.

### 2.1 End-to-end Fact-Checking Systems

ClaimBuster (Hassan et al., 2017) describes the process of automatic fact-checking as having several core components including a claim matching component, where claims are matched with previous fact checks if they exist, and a knowledge base lookup, which can directly answer questions using internal knowledge, e.g., Wolfram|Alpha. Automatic fact-checking systems, like this work, aim to fulfill the role of the knowledge base. More recent studies (Guo et al., 2022) have decomposed the process into claim detection, evidence retrieval, and verdict prediction and justification.

## 3 Fact-Checking Paradigm

We break down the task of fact-checking into three key steps: claim understanding, evidence retrieval, and claim-evidence alignment. Figure 1 provides an example of how the system processes the state-

[2] https://github.com/anon-naacl25/
[3] https://anon-naacl25.github.io/

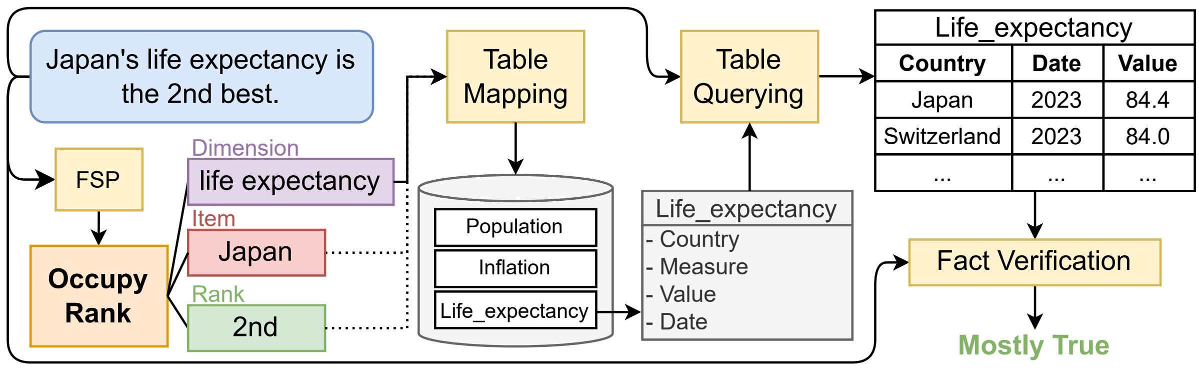


Figure 1: An example of our proposed paradigm. First, the frame and frame elements (FEs) are extracted using a frame-semantic parser (FSP). Then, FEs are mapped to a table which can then be queried to gather evidence for the claim. Finally, the evidence and claim are passed into a fact verification model to check the truthfulness of the claim.

ment "Japan's life expectancy is the 2nd best in Asia" to illustrate this process.

### 3.1 Claim Understanding

In the first step, the system focuses on understanding the specifics of the claim. Frame-semantic parsing enables the extraction of structured, predefined representations from claims. By identifying frames and their associated frame elements (FEs), the system gains a comprehensive understanding of the claim, which is critical for downstream tasks like evidence retrieval and verification. For example, in Figure 1, the Occupy_Rank frame is evoked, helping the system understand the claim in terms of the *Rank* of an *Item* along a *Dimension*.

Frame semantics not only facilitate claim understanding but also guide the fact-checking process by linking frame elements to appropriate data sources. While some frame elements, such as the *Agent* and *Issue* in the Vote frame, map directly to specific database tables, other frames may require a predicted mapping based on the text. Figure 1 illustrates an example of this predicted mapping.

### 3.2 Data Collection and Evidence Retrieval

The availability of trustworthy data is essential for fact-checking. Rather than relying on search engines that can return inconsistent results or be vulnerable to fake news injections (Huynh and Hardouin, 2023; Horne et al., 2019), our system stores ground-truth data in an internal database. Using data from reliable sources like the OECD and U.S. Congress ensures the integrity of the evidence.

Evidence retrieval is guided by the frame elements identified in the claim understanding phase. These frame elements act as filtering conditions for querying the appropriate tables or data sources. For example, in Figure 1, the Dimension frame element is used to query the relevant table for life expectancy statistics. By focusing on specific spans of the claim, this method minimizes confusion and enhances retrieval accuracy. Additionally, due to the fine-grained control enabled by frame-semantics, our proposed system can support unstructured data by adapting methods like retrieval-augmented generation (Lewis et al., 2020) using text chunking techniques to query large text corpora instead of a database.

### 3.3 Claim-Evidence Alignment

The final step involves aligning the retrieved evidence with the original claim. Frame elements play a critical role in ensuring that the evidence accurately reflects the claim's context. By only using specific spans of the claim, the system can align the claim and evidence more precisely. This approach also enhances explainability, as the system can point to the exact text used in the query.

Determining the truthfulness of a claim requires synthesizing retrieved evidence and understanding their connection to the claim as well as the intent behind the claim. For example, in Figure 1, the claim states that Japan has the 2nd best life expectancy, yet the data indicates that it is in fact the best. While this claim is not exactly true, the spirit of the claim is true, i.e., that the life expectancy is extremely high in Japan, and thus the model would predict that it is mostly true.

While fine-tuned approaches have shown slightly better performance on table-based fact-checking (Ye et al., 2023), we employ an LLM-based approach due to their ability to handle novel inputs without needing to retrain and for their ability to provide natural language explanations of predictions. Because this step is not a key focus of our work, we leave fact verification model-agnostic to allow future improvements. Thus, we can replace the current LLM with any fact verification model which takes structured evidence and a claim to perform fact verification.

## 4 Fact-Checking Case Studies

In this section we provide two case studies which utilize our proposed paradigm.

### 4.1 Voting Records

PolitiFact is a leading fact-checking organization that assesses the accuracy of public statements using verifiable evidence. Voting records [4] are a substantial portion of PolitiFact's fact-checks. In this case study, we focus on automatically fact-checking voting-related claims using the Vote frame and official U.S. congressional records. Specifically, we target the *Agent* and *Issue* Frame Elements (FEs), which refer to the voting entity and the topic of the vote, respectively.

**Datasets.** We compiled a dataset of official U.S. congressional voting records from the Congress Github Repository.[5] The dataset includes 342,466 bills from the 93rd Congress to the 117th Congress, with 7,195,798 individual votes across 22,447 roll calls. Each member of Congress cast an average of 4,230 votes from the 101st to the 117th Congress, and the dataset includes a total of 12,677 unique members of Congress since the 1st Congress.

In their fact-checking process, PolitiFact fact-checkers rely on evidence from official records and verified data to assess the truthfulness of claims. In the context of the Vote frame, claims often reference specific congressional bills. To evaluate our system, we constructed an evaluation dataset by extracting all PolitiFact fact-checks made before April 2022 that involve claims related to the Vote frame and reference at least one congressional bill. After manual verification, we collected 79 fact-checks, along with their corresponding bills, to form the evaluation set.

**Congress Member Identification.** To verify voting records, mapping the Agent FE to the correct member of Congress is essential. We use SQL queries to match congress members whose names are similar to the words in the Agent FE. In cases of name ambiguity, e.g., when two members share the same name, we default to the more recent member.

This stage presents several challenges. Claims often use nicknames, such as "Sleepy Joe" for Joe Biden or "Meatball Ron" for Ron DeSantis. To address this, we extracted two lists of political nicknames from Wikipedia (Wikipedia contributors, 2024a,b) to map nicknames to their corresponding congress members. Although not exhaustive, these lists should cover many common cases. Similarly, some members go by shortened or preferred names, such as "Joe" instead of Joseph or "Ted" for Rafael Edward Cruz. We handle this by utilizing a list of congress members' preferred names, supplemented by common alternatives [6] when necessary.

**Bill Matching.** Identifying the correct bill based on the extracted Issue FE is challenging because the Issue FE can refer to various types of information, such as abstract topics (e.g., "gun control"), specific actions or bills (e.g., the "Inflation Reduction Act of 2022"), or outcomes of legislation (e.g., "preventing women from getting abortions"). Additionally, bills often do not include the colloquial terms commonly used to describe them. For example, the "STOP School Violence Act of 2018" may be informally described as expanding access to guns in schools, even though the bill itself does not use this phrasing.

Because of these challenges, keyword-based search is insufficient for accurate evidence retrieval. To address this, we employ an asymmetric semantic similarity model, which allows us to identify bills that are semantically similar to the Issue FE, even when the language used in the claim and the bill differs significantly.

**Verifying alignment of claim to bill.** Determining whether a claim is refuted or supported by evidence presents several challenges. First, the system cannot rely solely on the vote (Yea or Nay) and the Position Frame Element (FE) (for or against). Claims may be made without a Position FE, and the relationship between the vote on a bill and the claim's Position FE can differ. For instance, a Yea vote on a bill may not indicate support for the claimed Issue. Consider the claim, "DeSantis voted against allowing abortions," in conjunction with a Yea vote on a bill that bans abortion. Here, the

[4] https://www.politifact.com/voting-record/
[5] https://github.com/unitedstates/congress
[6] https://github.com/carltonnorthern/nicknames

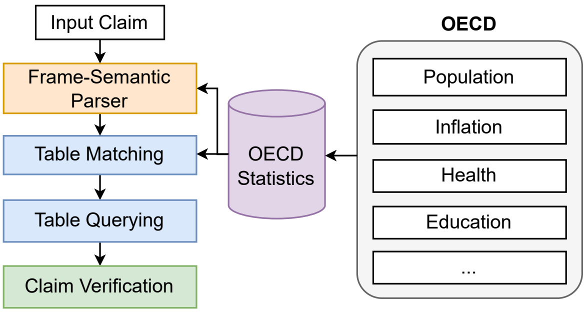


Figure 2: System design of OECD case study. The frame-semantic parser (orange) facilitates the claim understanding process, table matching and querying (blue) facilitate the evidence retrieval process, and the claim verification step (green) facilitates the fact verification.

vote supports the claim despite the discrepancy between the Position FE (against) and the vote on the bill (for/yea). Second, assessing whether a claim is supported or refuted by a bill vote requires an understanding of the bill itself and its implications.

**Claim Verification.** Finally, our system integrates the alignments between each bill and claim to perform fact verification over all of the retrieved evidence. We instruct a large language model (LLM) to conduct the final verification using the prompt defined in Appendix D.4.

### 4.2 OECD-Related Statistics

The OECD provides a wealth of trustworthy, observational data on a diverse set of statistics and serves as a strong focal point to explore a wider range of semantic frames. As a visual aid, we include a system overview in Figure 2.

**Dataset.** For the OECD Case Study, we collect all of the data tables available on the OECD Data Explorer. [7] We construct a SQLite database consisting of each data source, resulting in 434 tables with an average of 596,552 rows per table, totalling 4.1 billion cells.

**Relevant Table Identification.** To identify relevant tables for data extraction, we apply semantic similarity models to select the five most similar tables based on the extracted frame elements (FEs). The query consists of one or more FEs derived from frame-semantic parsing, while the target is a specific table. We encode each table's name and description as text for this purpose. Given the brevity of our queries and the length of the table descriptions, we utilize asymmetric semantic similarity models to ensure precise matches. To minimize the risk of overlooking relevant data, we retrieve the top five most similar tables.

[7] https://data-explorer.oecd.org/

**Querying Relevant Tables.** Querying the relevant tables presents challenges, as the system must recognize how values in the claim are represented in the database. In cases where direct mappings are available (e.g., in the Vote frame case study), we map FEs to specific tables. However, in the OECD case study, where no direct mapping exists between FEs and table columns, the system must predict this mapping. To handle these cases, we employ LLM-based code generation, allowing for the execution of multiple simple queries using programmed logic, rather than relying on complex text-to-SQL generation.

We represent the table schema as SQL code, following best practices from previous work (Gao et al., 2023), which studied different methods of representing SQL tables for text-to-SQL generation. Alongside the schema, we include representative example values from the database for each column. Encoded columns with fewer than five unique values include all values, while columns with more than 100 unique values are randomly sampled for ten representative values. For columns with a moderate number of unique values, we use a semantic similarity model to select the top ten most relevant values based on the claim. Given the large number of cells generated by this process, we use an LLM to refine the query and filter out irrelevant columns, focusing on those most critical for fact-checking the claim. The prompt for this LLM can be found in Appendix D.3.

**Fact Verification** In the final step, we used an LLM-based fact verification model. We represent the extracted evidence as a list of tab-separated values based on the data retrieved in the previous step. The model is instructed using the prompt in Appendix D.4 and outputs one of five verdicts: false, mostly false, half-true, mostly true, or true.

## 5 Experiments

### 5.1 Datasets

**Fact-checking Frames.** Arslan et al. (2020) introduced 11 manually defined semantic frames to extend the long-running Berkeley FrameNet project (Baker et al., 1998). Annotations for 936 sentences containing 1,029 frame-evoking targets and 3,570 frame elements are included in this

| Model | Frames | Frame Acc | FE Acc |
|---|---|---|---|
| Random | Vote | 0.488 | 0.254 |
| GPT-4o-mini | Vote | 0.974 | 0.618 |
| Vote FSP | Vote | **0.990** | **0.889** |
| Random | OECD | 0.602 | 0.000 |
| GPT-4o-mini | OECD | 0.537 | 0.372 |
| GPT-4o-mini* | OECD | 0.713 | 0.461 |
| OECD FSP | OECD | **0.742** | **0.873** |

Table 1: Performance of frame-semantic parsing model on fact-checking frames alongside simple baseline.

| Frame | Samples (%) |
|---|---|
| Taking_sides | 7,152 (34.0%) |
| Speech | 6,010 (28.6%) |
| Change_position_on_a_scale | 5,547 (26.4%) |
| Comparing_two_entities | 5,530 (26.3%) |
| Cause_change_of_position_on_a_scale | 4,675 (22.2%) |
| Vote | 3,229 (15.4%) |
| Comparing_at_two_different_points_in_time | 2,436 (11.6%) |
| Conditional_occurrence | 2,355 (11.2%) |
| Creating | 2,194 (10.4%) |
| Occupy_rank | 1,106 (5.3%) |
| Oppose_and_support_consistency | 1,010 (4.8%) |
| Recurrent_action_in_Frequency | 935 (4.4%) |
| Ratio | 932 (4.4%) |
| Capability | 869 (4.1%) |
| Occupy_rank_via_superlatives | 767 (3.6%) |
| Uniqueness_of_trait | 497 (2.4%) |
| Occupy_rank_via_ordinal_numbers | 329 (1.6%) |
| Recurring_action | 187 (0.9%) |
| None | 12 (0.1%) |

Table 2: Distribution of semantic frames in PolitiFact.

| Model | Data | R@K |
|---|---|---|
| `BM25` | OECD | 0.323 |
| `distilbert-tas-b` | OECD | 0.505 |
| `text-embedding-3-large` | OECD | 0.610 |
| `roberta-base-v2` | OECD | **0.726** |
| `text-embedding-3-large` | Vote | 0.032 |
| `stella_en_1.5B_v5` | Vote | 0.144 |
| `distilbert-tas-b` | Vote | **0.165** |

Table 3: Performance comparison of different semantic similarity models on GPT-extracted topics and extracted frame elements. K=10 for Vote and K=5 for OECD.

dataset along with the newly-defined frames. These frames can be found in Appendix 6.

**PolitiFact Fact-checks** For our analysis, we utilized a dataset of 21,024 PolitiFact fact-check articles collected as of April 2022. Each article contains a claim, a detailed fact-check, a verdict, and a list of sources. To focus on voting-related claims, we extracted 1,552 (7.4%) fact-checks that mention some form of "vote." From this subset, we identified 79 fact-checks that cite congress.gov and evoke the Vote frame. For each voting claim, we manually verified the bills referenced within the fact-check are related to the claim to ensure the accuracy of our dataset.

Additionally, we collected 68 articles that cite oecd.org for our analysis of OECD-related claims. Each OECD claim was manually verified and mapped to the statistics which can be used to fact-check it. These two sets of claim-evidence pairs—the voting-related claims and the OECD-related claims—serve as evaluation sets for our case studies.

### 5.2 Frame-Semantic Parsing

For frame-semantic parsing, we combine the frame identification model developed by Devasier et al. (2024) with a frame element identification model based on AGED (Zheng et al., 2023). By combining these state-of-the-art approaches, we unify frame and frame element identification into a single model. This model is then trained only on the fact checking frames introduced in Arslan et al. (2020).

To evaluate the performance of our model for the case studies, we compare it with a GPT-4o-mini model using OpenAI's structured generation[8]. We evaluate each model using accuracy on frame identification and exact match accuracy on FE identification in Table 1. We define a separate prompt for each case study in Appendix D.1.

[8]At the time of submission, GPT-4o-mini is the only model which supports structured generation.

### 5.3 Claim Coverage

One concern which may arise is regarding the applicability of our system for assisting in the fact checking process. To understand this, we aim to identify the coverage of a frame-based approach towards automatically fact-checking claims. Due to the lack of annotations for all of the fact-checking frames in Arslan et al. (2020), we utilize a zero-shot GPT-4o-mini model to identify the frames evoked in the claim of each PolitiFact fact-check (Table 2). Claims which do not evoke a frame studied in this work are categorize as none. The prompt used for this model can be found in Appendix D.1.

### 5.4 Evidence Retrieval

We study the capability of our system to retrieve the relevant information which can be used to fact-check the claim. To do this, we found the percent-

| Model | Query | Data | R@K |
|---|---|---|---|
| `RoBERTa (v2)` | Full claim | OECD | 0.653 |
| `RoBERTa (v2)` | FE | OECD | **0.726** |
| `Max Possible` | - | OECD | *0.910* |
| `distilbert-tas-b` | Full claim | Vote | 0.143 |
| `distilbert-tas-b` | Issue FE | Vote | **0.165** |
| `Max Possible` | - | Vote | *0.568* |

Table 4: Performance of different methods of representing the query in the OECD case study's table matching. OECD and Vote claims use Recall@5/10, respectively.

| Model | Dataset | Accuracy |
|---|---|---|
| GPT-4o Naive | Vote | 0.044 |
| Our w/ Irrelevant | Vote | 0.076 |
| Our w/o Irrelevant | Vote | <u>0.207</u> |
| Our w/ Irrelevant $^{\pm}$ | Vote | 0.253 |
| GPT-4o Naive $^{\pm}$ | Vote | 0.324 |
| Our w/o Irrelevant $^{\pm}$ | Vote | **0.690** |
| GPT-4o Naive | OECD | 0.073 |
| Our w/ Irrelevant | OECD | 0.214 |
| Our w/o Irrelevant | OECD | <u>0.429</u> |
| GPT-4o Naive $^{\pm}$ | OECD | 0.537 |
| Our w/ Irrelevant $^{\pm}$ | OECD | 0.607 |
| Our w/o Irrelevant $^{\pm}$ | OECD | **0.821** |

Table 5: Fact verification performance on different case studies. $^{\pm}$ indicates the off-by-one performance. Bolded values indicate the best-performing settings overall and underlined values are only considering exact matches.

age of bills and statistics available in our database for voting and OECD-related claims, respectively. This serves as a performance ceiling on our collected claims as our system would be unable to fact-check claims which do not have available data. These values are found in Table 4 (Max Possible).

For semantic similarity, we compare commonly used embedding models (Table 3), namely RoBERTa (Liu et al., 2019) and DistilBERT-TAS-B (Hofstätter et al., 2021). Additionally, we compare with OpenAI's text-embedding-3-large model and provide a baseline BM25 approach. We also experiment with different approaches of representing the query used to find relevant data. In Table 4, we compare the performance of using the entire claim with only the relevant FE. To evaluate the performance of these queries, we calculate the recall@K, where K=5 for OECD claims and K=10 for Vote claims. The relevant FEs used for each frame can be found in Appendix A.

### 5.5 Fact Verification

Finally, we study the ability of our fact verification system to fact-check claims using structured data by comparing the verdicts presented in each PolitiFact fact-check to the model's predicted verdict. We have replaced PolitiFact's "pants on fire" verdicts with a false verdict. To determine whether the LLM is relying on internal knowledge or if it is using the provided data, we evaluate the performance of GPT-4o without any data. We measure accuracy as the primary evaluation metric and also calculate off-by-one accuracy, where, for example, a prediction of "mostly false" is considered correct if the label is "false." The results of these experiments are summarized in Table 5.

## 6 Results

### 6.1 Frame-Semantic Parsing

We evaluated the frame-semantic parsing model on fact-checking frames introduced by Arslan et al. (2020), comparing it against GPT-4o-mini due to its structured output capability. Table 1 shows the performance on both the Vote and OECD frames. We found that the GPT-4o-mini model tends to over-predict the number of frames in a sentence. When we only use the first predicted frame, the performance is increased significantly. We saw similar cases with frame element predictions where GPT-4o-mini tends to predict more frame elements than actually exist in the input. We include examples of each of these cases in Appendix C.2.

### 6.2 PolitiFact Survey

Similar to Section 6.1, we found that GPT-4o-mini tends to predict more frames than there may actually be. We believe this is likely due to the annotations from Arslan et al. (2020) being sampled from PolitiFact and likely having a similar structure. We found the annotated samples evoked 1.1 frames on average while GPT-4o-mini predicts an average of 2.1 frames. Another possible explanation is that the model is able to identify frames for lexical units–words which evoke frames–which are not defined.

One indication of this is on the Vote frame, which only has a lexical unit defined for *vote.v*; however, it is possible to evoke the Vote frame using other words. For example, the statement "I passed a bill" implies that a vote of affirmation was cast on that bill. We found only 1,198 (5.7%) PolitiFact fact-checks which mention "vote" while the model predicts that almost 3 times as many evoke the Vote frame. While we found many claims which evoke a Vote frame without an explicit mention of the word "vote", further analysis is needed to get a more accurate understanding of the true distribution of frames evoked by factual claims.

### 6.3 Evidence Retrieval

We found that embedding models' performance is task-specific and there is no single-best embedding model for every task. For example, RoBERTa performed the best for finding the right OECD table while DistilBERT-TAS-B was better for matching claims to their corresponding bills. We also found that, in general, using the claim-specific frame element performs better than using the entire claim as a query, as shown in Table 4.

For voting-related claims, we found that identifying the bill referenced in a particular claim is very difficult. Our best performing model, DistilBERT-TAS-B, was only able to obtain a Recall@10 of 0.165. For OECD-related claims, identifying the table to use to fact-check the given claim was much easier, with the best model, RoBERTa, achieving a recall@5 of 0.726. This is likely because the similarity between the relevant FE and the table names were better-represented by semantic similarity models. This indicates a potential area of significant improvement in the methodology and representation of bill texts according to their topics and implications, in addition to the text itself.

### 6.4 Fact Verification

On OECD claims, 62% of the generated queries were able to retrieve relevant data while only 36% of the voting claims were able to retrieve relevant bills. These values put an upper bound on the capabilities of our fact verification model as without evidence, the model will not be able to provide useful fact-checks. In general, we found that language models tend not to predict a claim to be true based on the given data, yet they have no issue with predicting a claim to be false, and often do so. We believe this is analogous to proofs, it is quite easy to prove by contradiction, if you find a single vote or country statistic which contradicts the claim, while it is much harder to assert that a claim is true without looking at all of the data.

We found that on Vote claims, a naive GPT-4o correctly predicted the truthfulness of the factual claim with an accuracy of 0.286; however, when voting data is added, the performance drops to 0.076. This suggests that, while there may be some internal knowledge on certain people and bills, it is unlikely that the model is relying on it when given specific voting records. This also indicates that the prompt given to the model is sufficient in preventing the LLM from relying on internal knowledge.

When excluding claims for which data was not found, we observed absolute performance increases of +13.1% and +21.5% for Vote and OECD claims, respectively. We also found that, while the exact-match accuracy of the system is low, there is a strong correlation between predictions and labels, as indicated by the off-by-one accuracy in Table 5. We include example outputs for the explanations of these models in Appendix C.1.

## 7 Conclusion

This research contributes to the evolving field of automatic fact-checking by proposing a novel paradigm that incorporates frame semantics to enhance the evidence retrieval process and enable fine-grained control of downstream tasks. By developing and presenting a pilot dataset specifically annotated for real-world claims, this work helps to address the underexplored challenge of fact-checking using massive, structured data sources.

Our findings demonstrate the promise of frame-semantic parsing in improving evidence retrieval for automatic fact-checking. In both case studies, we observed that using frame elements extracted from claims instead of entire claims led to substantial improvements in recall metrics, highlighting the utility of frame-based understanding in identifying relevant evidence. These results suggest that semantic frames not only offer a structured and explainable approach to parsing claims but also guide retrieval models more effectively toward the correct information within large datasets.

Looking ahead, the integration of frame semantics into fact-checking systems holds promise for further advancements in both accuracy and interpretability. Future work could explore broader applications of this approach, including expanding the dataset, refining the models, and incorporating additional semantic frames to cover more claims.

## Limitations

Despite the advancements made in this study, our system's limitations should be acknowledged. While this study focuses primarily on improving evidence retrieval and frame-semantic parsing, it does not comprehensively evaluate different fact verification approaches. Fact verification is critical for automatic fact-checking, and a deeper investigation into various verification strategies, including fine-tuned models and LLM-based approaches, could further enhance system performance.

Another limitation is that our system's effectiveness is constrained by the availability of reliable, manually curated data. For claims that lack relevant data in our databases (e.g., OECD tables or U.S. congressional records), the system cannot retrieve evidence, which limits its fact-checking capabilities. This limitation demonstrates the importance of expanding data sources and improving coverage in future iterations of the system.

## Ethics and Risks

Automated fact-checking systems, such as the one presented in this work, bring both opportunities and ethical challenges. One key concern is the potential spread of misinformation due to model limitations or errors. Users may over-rely on AI verdicts, leading to the amplification of false positives or negatives, especially in politically sensitive contexts. Biases present in both the models (e.g., GPT-4o-mini, RoBERTa) and the datasets (e.g., PolitiFact, OECD data) could skew fact-checking outcomes, favoring certain political narratives or ideologies.

Another concern is the transparency and accountability of AI systems. If users are unaware of how models arrive at their conclusions, it may be difficult to hold these systems accountable for erroneous outcomes. This opacity could diminish trust in both the fact-checking tool and the institutions that deploy it. Moreover, the collection and processing of political data raise potential privacy concerns, especially regarding the use of public records in ways individuals may not expect.

Marginalized communities may also be disproportionately affected by such systems, as they might misinterpret claims relevant to those groups or lack sufficient representation in training data. Similarly, the system could be exploited by adversaries who craft ambiguous or misleading claims designed to confuse AI, leading to manipulated fact-checks.

Lastly, the over-dependence on specific sources like PolitiFact or congress.gov could limit the tool's scope. Ethical development of such systems requires attention to these risks to ensure fairness, accuracy, and social responsibility in their use.

## References


Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. 2022. PubHealthTab: A public health table-based dataset for evidence-based fact checking. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1–16, Seattle, United States. Association for Computational Linguistics.

Rami Aly, Zhijiang Guo, M. Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. Feverous: Fact extraction and verification over unstructured and structured information. *ArXiv*, abs/2106.05707.

Fatma Arslan, Josue Caraballo, Damian Jimenez, and Chengkai Li. 2020. Modeling factual claims with semantic frames. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2511–2520, Marseille, France. European Language Resources Association.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Mostafa Bouziane, Hugo Perrin, Amine Sadeq, Thanh Nguyen, Aurélien Cluzeau, and Julien Mardas. 2021. FaBULOUS: Fact-checking based on understanding of language over unstructured and structured information. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 31–39, Dominican Republic. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, Suchir Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir R. Radev, Marilyn Ostendorf, Luke S. Zettlemoyer, Noah A. Smith, and Tao Yu. 2022. Binding language models in symbolic languages. *ArXiv*, abs/2210.02875.

Tsun-Hin Cheung and Kin-Man Lam. 2023a. Factllama: Optimizing instruction-following language models with external knowledge for automated fact-checking. *Preprint*, arXiv:2309.00240.

Tsunhin Cheung and Kin Man Lam. 2023b. Factllama: Optimizing instruction-following language models with external knowledge for automated fact-checking. *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 846–853.

Jacob Devasier, Yogesh Gurjar, and Chengkai Li. 2024. Robust frame-semantic models with lexical unit trees and negative samples. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6930–6941, Bangkok, Thailand. Association for Computational Linguistics.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2023. Text-to-sql empowered by large language models: A benchmark evaluation. *Preprint*, arXiv:2308.15363.

Zihui Gu, Ju Fan, Nan Tang, Preslav Nakov, Xiaoman Zhao, and Xiaoyong Du. 2022. PASTA: Table-operations aware fact verification via sentence-table cloze pre-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4971–4983, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017. Claimbuster: the first-ever end-to-end fact-checking system. *Proc. VLDB Endow.*, 10(12):1945–1948.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 113–122, New York, NY, USA. Association for Computing Machinery.

Benjamin D. Horne, Jeppe Nørregaard, and Sibel Adali. 2019. Robust fake news detection over time and attack. *ACM Trans. Intell. Syst. Technol.*, 11(1).

Daniel Huynh and Jade Hardouin. 2023. Poisongpt: How we hid a lobotomized llm on hugging face to spread fake news. Accessed: 2024-10-09.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Seattle, United States. Association for Computational Linguistics.

Saehan Jo, Immanuel Trummer, Weicheng Yu, Xuezhi Wang, Cong Yu, Daniel Liu, and Niyati Mehta. 2019. Verifying text summaries of relational data sets. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 299–316, New York, NY, USA. Association for Computing Machinery.

Georgios Karagiannis, Mohammed Saeed, Paolo Papotti, and Immanuel Trummer. 2020. Scrutinizer: fact checking statistical claims. *Proc. VLDB Endow.*, 13(12):2965–2968.

Mian Abdul Khaliq, P. Chang, M. Ma, B. Pflugfelder, and F. Mileti'c. 2024. Ragar, your falsehood radar: Rag-augmented reasoning for political fact-checking using multimodal large language models. *ArXiv*, abs/2404.12065.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *ArXiv*.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L Petruck, Christopher R. Johnson, Collin F. Baker, and Jan Scheffczyk. 2016. FrameNet II: Extended theory and practice. *International Computer Science Institute, Berkeley, California.*

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a known lie:

Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online. Association for Computational Linguistics.

Ronit Singhal, Pransh Patwa, Parth Patwa, Aman Chadha, and Amitava Das. 2024a. Evidence-backed fact checking using rag and few-shot in-context learning with llms. *ArXiv*, abs/2408.12060.

Ronit Singhal, Pransh Patwa, Parth Patwa, Aman Chadha, and Amitava Das. 2024b. Evidence-backed fact checking using rag and few-shot in-context learning with llms. *Preprint*, arXiv:2408.12060.

Nancy X. R. Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021. SemEval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (SEM-TAB-FACTS). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 317–326, Online. Association for Computational Linguistics.

William Yang Wang. 2017. “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Yuxia Wang, Minghan Wang, Hasan Iqbal, Georgi Georgiev, Jiahui Geng, and Preslav Nakov. 2024. Openfactcheck: A unified framework for factuality evaluation of llms. *Preprint*, arXiv:2405.05583.

Wikipedia contributors. 2024a. List of nicknames of presidents of the united states — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=List_of_nicknames_of_presidents_of_the_United_States&oldid=1226749824. [Online; accessed 3-June-2024].

Wikipedia contributors. 2024b. List of nicknames used by donald trump about other people — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=List_of_nicknames_used_by_Donald_Trump_about_other_people&oldid=1226728769. [Online; accessed 3-June-2024].

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’23, page 174–184, New York, NY, USA. Association for Computing Machinery.

Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. 2022. ReasTAP: Injecting table reasoning skills during pre-training via synthetic reasoning examples. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9006–9018, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ce Zheng, Yiming Wang, and Baobao Chang. 2023. Query your model with definitions in framenet: an effective method for frame semantic role labeling. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI’23/IAAI’23/EAAI’23. AAAI Press.

# A Studied Frames

Table 6 provides a comprehensive list of frames studied in this work along with the frame elements used to predict table mappings.

# B Reproducibility

We trained our frame-semantic parser using a NVIDIA GTX 1080 Ti in about 6 hours on a single run without any hyperparameter tuning. Github Copilot was used in the development of some of our code.

## B.1 Software License

Our source code and data are released with the MIT License.

## B.2 Models

Unless stated otherwise, all models used are the “base” variant. All semantic similarity models are available via the sentence-transformers library.[9] Table 7 contains all of the models used and their details.

# C Examples

In this section, we provide a few example outputs of our system to give a better understanding of the outputs of our models.

## C.1 Vote-Claim Alignment

We include two sample claims and their corresponding top three retrieved bills along with the vote-claim alignment for each bill in Table 9.

## C.2 Fact Verification Predictions

**Vote** We include two claims and their corresponding predictions based on the retrieved bills in Table 9.

[9] https://huggingface.co/sentence-transformers

| Frame | Frame Element | Frame Definition |
|---|---|---|
| Occupy_rank | Dimension | Items occupying a certain Rank within a hierarchy. |
| Occupy_rank_via_superlatives | Dimension | An Item occupying a Rank specified by a superlative. |
| Comparing_two_entities | Comparison_criterion | Comparing two entities using a Comparison_criterion and qualifying with a Degree. |
| Comparing_at_two_different_ points_in_time | Comparison_criterion | Comparing an Entity with itself at two different points in time using a Comparison_criterion and qualifying with a Degree. |
| Cause_change_of_position_ on_a_scale | Item | Words that indicate an Agent or Cause affects the position of an Item on a scale. |
| Capability | Event | An Entity meets the pre-conditions for participating in an Event. |

Table 6: Frames used for OECD Case Study, along with the Frame Elements extracted from corresponding Frames.

| Model | Parameters | URL |
|---|---|---|
| msmarco-distilbert-base-tas-b | 66M | msmarco-distilbert-base-tas-b |
| msmarco-roberta-base-v2 | 125M | msmarco-roberta-base-v2 |
| msmarco-roberta-base-v3 | 125M | msmarco-roberta-base-v3 |
| stella_en_1.5B_v5 | 1.5B | stella_en_1.5B_v5 |
| text-embedding-3-large | unknown | text-embedding-3-large |

Table 7: Models and Their Corresponding Hugging Face URLs

| Claim | Predicted Verdict | Explanation | Fact Check |
|---|---|---|---|
| Americans work way more than an average of industrialized countries around the world. | Mostly True | The data shows that the United States has an annual average of 1829 hours worked per person, which is higher than many other industrialized countries in the dataset, such as Germany (1400.4 hours), France (1518.0 hours), and Japan (1729.0 hours). However, there are some countries like Mexico (2229.0 hours), Korea (2076.0 hours), and Chile (1994.0 hours) with higher working hours than the U.S. This indicates that while Americans work more than many other industrialized countries, they do not work more than all. The claim is mostly true but lacks the nuance that there are industrialized countries where people work more hours. | PolitiFact |
| The United States is the ONLY industrialized country without universal healthcare. | True | The retrieved data indicates that countries such as Italy, the Netherlands, Spain, Canada, Korea, Australia, France, Germany, the United Kingdom, and Sweden all have near-universal health insurance coverage through government or compulsory schemes, with coverage rates often very close to 100% of the population. In contrast, the United States has only 38.1% of its population covered by government or compulsory health insurance, suggesting that it lacks universal healthcare coverage. This supports the claim that the United States is the only industrialized nation without such a system based on the data provided. | PolitiFact |

Table 8: OECD fact check example predictions and explanations

| Claim | Bill Title | Vote | Alignment | Prediction |
|---|---|---|---|---|
| Marco Rubio voted against the bipartisan Violence Against Women Act | 112 S 1925 Violence Against Women Reauthorization Act of 2012 | Nay | Supports | True |
| | 17 HR 3233 National Commission to Investigate the January 6 Attack on the United States Capitol Complex Act | Nay | Irrelevant | |
| | 117 HR 350 Domestic Terrorism Prevention Act of 2022 | Nay | Irrelevant | |
| Chuck Grassley was voting to slash Medicare when voting against the debt ceiling bill | 117 S 610 Protecting Medicare and American Farmers from Sequester Cuts Act | Nay | Supports | Mostly False |
| | 117 S 1301 Promoting Physical Activity for Americans Act | Nay | Irrelevant | |
| | 17 HR 1868 To prevent across-the-board direct spending cuts, and for other purposes | Yea | Refutes | |

Table 9: Examples of voting-related claims with the corresponding retrieved bills, votes on the bill, the vote-claim alignment, and fact verification prediction.

**OECD** We include two predictions and their explanations on OECD claims in Table 8.

# D LLM Prompts

## D.1 Frame-Semantic Parsing

Listing 1: Frame-semantic parser for the Vote frame.

```
Identify if a 'Vote' semantic frame is
    evoked in a given sentence. If it is
    , extract and list the relevant
    frame elements associated with the
    voting event.

A 'Vote' frame is defined as: An Agent
    makes a voting decision on an Issue.

Frame elements to identify:
- Agent: The conscious entity (usually a
     person) executing the voting
    decision.
- Issue: The subject or matter that the
    Agent is voting on with a particular
     position.
- Side: Additional person(s) involved in
     the voting on the same Issue
    alongside the Agent.
- Position: The stance the Agent takes,
    expressing whether their vote is in
    favor or against the Issue.
- Frequency: How often the Agent has
    made this voting decision regarding
    the Issue.
- Time: When the Agent performed the
    voting decision.
- Place: Location where the vote took
    place.
- Support_rate: The percentage of the
    Agent's total votes aligning with
    the Side over a set period.

# Notes

- If a frame element is not mentioned in
     the sentence, use "" for its value.
- Carefully distinguish between elements
    ; some may have overlapping
    characteristics.
- Consider synonyms or variations of
    terms related to voting when
    analyzing the sentence.
- Frame elements should quote exactly
    from the input
```

Listing 2: Frame-semantic parser for OECD frames.

```
Identify semantic frames evoked by a
    given factual claim and extract
    relevant frame elements for each
    identified frame.

Consider the predefined semantic frames
    and their elements:

- Occupy_rank_via_superlatives:
  - Item: Entity occupying the rank.
  - Rank: Rank held, often defined by a
      superlative.
  - Dimension: Aspect along which the
      ranking occurs (e.g., speed, age).
  - Comparison_set: Group of entities
      being compared.
  - Time: Time period when the item
      occupies the rank.

- Occupy_rank:
  - Item: Entity occupying the rank.
  - Rank: Rank held.
  - Dimension: Criterion used for
      ranking.
  - Comparison_set: Set of entities
      being ranked.
  - Time: Time period when the item
      holds the rank.
```

```
- Change_position_on_a_scale:
 - Item: Entity whose position on a
    scale changes.
 - Attribute: Property or scale of the
    change.
 - Difference: Amount of change.
 - Final_value: Final position on the
    scale.
 - Initial_value: Initial position on
    the scale.
 - Path: Progression between positions
    on the scale.
 - Speed: Rate of change.
 - Correlated_variable: Related variable
     that changes with the Attribute.
 - Manner: Method of performing the
    action.
 - Degree: Extent to which the change
    occurs.
 - Circumstances: Context or conditions
    for the change.
 - Result: Outcome of the change.
 - Group: Collection of Items undergoing
     change.
 - Time: Time frame during which the
    change occurs.
 - Duration: Length of time for the
    change to happen.
 - Value_range: Range of values along
    the scale.
 - Initial_correlate: State
    corresponding to the Initial_value.
 - Final_correlate: State corresponding
    to the Final_value.
 - Place: Location where the Attribute
    is measured.
 - Initial_state: State of the Item
    before the change.
 - Final_state: State of the Item after
    the change.
 - Period_of_iterations: Duration from
    start to stop of repeated changes.
 - Particular_iteration: Specific
    instance of a repeated event.
 - Containing_event: Broader event
    encompassing the change.
 - Explanation: Rationale for the change
    .

- Comparing_two_entities:
  - Entity_1: First entity in comparison
    .
  - Entity_2: Entity serving as the
     comparison point.
  - Comparison_criterion: Criterion for
     comparison.
  - Degree: Extent of the comparison.
  - Time: Period over which the
     comparison takes place.

- Comparing_at_two_different_
   points_in_time:
  - Entity: Entity compared to itself at
      different times.
  - First_point_in_time: First time
     period in comparison.
  - Second_point_in_time: Second time
     period in comparison.
  - Comparison_criterion: Criterion
      being compared.
  - Degree: Extent of the difference.
  - Difference: Magnitude of change
      between the two times.

# Steps

1. Carefully read and analyze the
    factual claim.
2. Determine which semantic frame(s) are
     invoked by the claim.
3. For each evoked frame, extract the
    appropriate frame elements directly
    from the claim.
4. Document each evoked frame and the
    extracted elements.

# Output Format

- List each evoked frame followed by its
     extracted elements in a structured
    form.
- Ensure all extracted frame elements
    are clearly labeled and match the
    exact inputs.

# Notes
- Some claims may trigger multiple
    frames; ensure all applicable frames
     are evaluated.
- If specific frame elements cannot be
    determined, note their absence for
    completeness.
- If a frame element does not explicitly
     appear in the claim, leave it blank
    .
```

## D.2 Claim-Bill Alignment

Listing 3: Claim-Bill Alignment

```
Given the following factual claim, bill
    summary, and vote on the bill,
    evaluate whether the content of the
    bill summary and the voting record
    align with the given claim. You may
    consider factors such as the main
    objectives of the bill and
    unintended or implicit consequences.
     Your task is to determine if the
    information provided in the bill
    summary and the voting record
    supports or refutes the given
    factual claim. Return your
    explanation and one of the following
     labels in JSON format.

Bill Summary:
{summary}

Vote: {vote_type}
Claim: {claim}

Labels:
Supports - The vote on this bill
    directly or indirectly supports the
    claim.
Refutes - The vote on this bill
    explicitly refutes the claim.
```

```
Inconclusive - The vote on this bill
    does not provide enough information
    to support or refute the claim.
Irrelevant - The vote on this bill is
    not relevant to the claim at all.
```

### D.3 OECD Data Query

Listing 4: Retrieve Data for Fact-Checking Claim

```
Your task is to write a Python function
    named retrieve_data() that retrieves
     data for fact-checking the claim:
{claim}

Given the following database schemas
    from OECD_Data.db:
{table_descriptions}

Instructions:
- Use LIKE only for textual columns. For
     numerical columns (e.g., year,
    value), use appropriate comparison
    operators like = or <=.
- Do not modify the database.
- The function must contain all
    necessary imports inside it and take
     no parameters, though passing
    parameters (e.g., file paths or
    claim details) should be considered
    in future iterations.
- Return the data in pandas DataFrames.
    Multiple DataFrames can be returned
    as a list if needed.
- If no relevant data is found, return '
    Data is not Available'.
- Use only actual values found in the
    columns. If aggregation is possible
    (e.g., summing or averaging
    categories), return aggregated
    results, unless the claim specifies
    otherwise. Use appropriate
    aggregation methods based on the
    data.
- Always return relevant columns (avoid
    '*' when possible). Select the
    necessary columns based on the claim
    .
- Treat 'we' in the claim as referring
    to 'United States', but consider the
     context of the claim for potential
    exceptions.
- If any 'unit_of_measure' is 'National
    currency', use the 'USD_value'
    column instead of 'value'.
- Use the nearest available date based
    on the provided dictionary: {
    nearest_dates}. If no close date is
    found, return 'Data is not Available
    '.
- For multiple tables, create multiple
    queries as needed. If combining data
     from multiple tables is required,
    ensure consistency in merging and
    handling differences in structure.
- Do not filter by 'country' or '
    unit_of_measure'; that will be
    handled later.

Output:
- Return a list of pandas DataFrames or
    'Data is not Available' if no
    relevant data is found.
```

Listing 5: Clean Data for Fact-Checking Claim

```
Your task is to write a Python code
    function named `clean_data()` to
    filter and clean data for fact-
    checking the claim {claim} from `
    OECD_Data.db`.

Extracted dataframes from the previous
    step:
{list_of_dfs}

These dataframes were generated using
    this code:
{code}

The schema for these dataframes is:
{schema_str}

Notes:
- The function must have all imports
    inside it and take no parameters.
- This function will run independently
    of the given code, meaning you need
    to re-extract the data from the
    database to clean it for fact-
    checking the claim:
  {claim}
- Do not modify the original way the
    data was extracted unless the input
    code is very incorrect; just add
    more filters/conditions based on the
     claim.
- Keep the same aggregated structure
    from the original extraction code
    unless the claim specifies otherwise
    .
- Filter by the nearest date: {
    nearest_date}.
- Exclude tables only if they are
    irrelevant to the claim.
- Use only the actual tables and data
    that were extracted; do not make up
    new tables.
- Add more filters to narrow down to
    claim-relevant metrics; try to have
    one value per non-numeric column
    wherever possible.
- Claim-relevant metrics may appear in
    columns other than `'measure'`, so
    reference the claim to identify the
    appropriate columns to filter by.
- Use the retrieval code to identify
    useful information for your filters,
     if needed.
- When filtering by `'unit_of_measure'`,
     ensure to use a standardized metric
     (e.g., a common currency) to
    simplify further analysis.

Output:
- A list containing cleaned pandas
    DataFrames based on the claim. If
    the code is run and the data is not
    available, return `'Data is not
```

```
    Available'`.
```

### D.4 Fact Verification

Listing 6: Vote - Verify Claim and Provide Verdict

```
You are given the following claim, and 5
     bills along with the bill title, a
    short summary of the bill, the vote
    cast on the bill and the alignment
    of the bill with the claim.
Alignment here means whether an
    individual bill supports, refines,
    or is irrelevant to the claim.
Your task is to give one of the given
    fact-check labels to the claim based
     on the evidence which are the bills
    .
You may consider factors such as the
    main objectives of the bill and
    unintended or implicit consequences.
Return your explanation and one of the
    following labels in JSON format.

Claim: {claim}

Bill Title 1: {bill_title_1}
Bill Summary 1: {bill_summary_1}
Vote 1: {vote_type_1}
Alignment1: {alignment_1}

[...]

Bill Title N: {bill_title_N}
Bill Summary N: {bill_summary_N}
Vote N: {vote_type_N}
Alignment N: {alignment_N}

Labels:
True: The given bills support the claim.
MostlyTrue: The given bills mostly
    support the claim.
HalfTrue: The given bills can only
    partly support or refute claim.
MostlyFalse: The given bills mostly
    refute the claim.
False: The given bills refute the claim.
Irrelevant: The given bills are not
    relevant to the claim.

Return the JSON object with the label
    and the explanation. The fields
    should be 'Label' and 'Explanation'.
```

Listing 7: OECD - Verify Claim and Provide Verdict

```
Your task is to verify the claim using
    the retrieved data and provide a
    verdict.
Claim: {claim}

Retrieved Data:
{formatted_data}

Instructions:
- Use only the provided data, regardless
     of its perceived relevance, to
    analyze the claim.
- The verdict must be based solely on
    the retrieved data. Do not rely on
    external knowledge.
- Ensure you consider the spirit of the
    claim in the fact-check and not just
     the precise verbage.

Verdict Categories:
- True: The statement is fully accurate,
     with no significant information
    missing.
- Mostly True: The statement is accurate
     but requires clarification or
    additional context.
- Half-True: The statement is partially
    accurate but omits important details
     or misrepresents the context.
- Mostly False: The statement contains
    some truth but overlooks key facts
    that would significantly alter the
    impression.
- False: The statement is completely
    inaccurate.

Output:
- Provide a verdict in the format:

Verdict: [False, Mostly False, Half-True
    , Mostly True, True]; Explanation: [
    Your reasoning].
```