Enabling Automated Fact Checking of Voting Related Claims Using Frame Semantic Parsing and Semantic Search

Rishabh Mediratta
rxm5684@mavs.uta.edu
University of Texas at Arlington
Arlington, Texas, USA

Jacob Devasier
jacob.devasier@mavs.uta.edu
University of Texas at Arlington
Arlington, Texas, USA

Chengkai Li
cli@uta.edu
University of Texas at Arlington
Arlington, Texas, USA

ABSTRACT

We introduce a transparent and automated fact-checking system focused on verifying voting-related factual claims. Current fact-checking methods frequently suffer from a lack of transparency, which hinders users' ability to trust and comprehend the underlying rationale behind the results. The opaque nature of these solutions can leave users questioning the reliability and impartiality of the fact-checking process, as they are unable to fully grasp the methodology and evidence used to arrive at the conclusions. In this paper, we introduce a new approach to fact verification that utilizes frame-semantic parsing to give structured and interpret-able fact-checking and semantic search to find evidence to support the verification of voting-related claims. By concentrating on claims related to voting, we can leverage publicly accessible voting records from official United States congressional sources and employ frame semantic parsing to extract pertinent information from these claims.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; **Lexical semantics**; **Information extraction**.

KEYWORDS

automated fact-checking, semantic search, frame semantic parsing, explainability

ACM Reference Format:
Rishabh Mediratta, Jacob Devasier, and Chengkai Li. 2024. Enabling Automated Fact Checking of Voting Related Claims Using Frame Semantic Parsing and Semantic Search. In *Proceedings of KDD Undergraduate Consortium 2024 (KDD-UC '24).* ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

The rapid spread of misinformation and disinformation in today's digital world highlights the urgent need for effective fact-checking solutions. Automated fact-checking methods have shown potential in curbing false information, but most current systems face a significant challenge: they lack transparency and explain-ability. Many rely on machine learning models, which are often complex and difficult to interpret, making their decision-making processes opaque. This opacity can erode trust, as users struggle to understand the reasoning behind fact-checking results. This issue is particularly worrisome in high-stakes fields like journalism, healthcare, and finance, where the credibility of fact-checking outcomes is crucial.

Recent efforts to automate fact-checking have primarily focused on fake news/misinformation detection [6, 14] and fact verification [1, 12]. Fake news detection generally involves identifying news pieces containing nonfactual information intended to deceive the consumer or misrepresent truthful information [7]. The other important approach to mitigate misinformation which we focus on this work is fact verification. Fact verification seeks to determine whether a claim is true or false based on a given piece of evidence [15]. We address two primary challenges of fact verification in this work:

- **Lack of Evidence**: Fact verification often assumes the availability of evidence that can confirm or refute a given claim, which is not always the case for real-world statements. Finding evidence for any general claim is extremely challenging given that the domain of evidence required is extremely large. Hence, in this work, we focus on fact-checking voting-related claims, leveraging publicly available voting records from the U.S. Congress to pinpoint the evidence.
- **Transparency in Fact-Checking**: For fact-checking to be effective, it is essential that the public trusts the outcomes. To achieve this, people need to understand how fact-checking models arrive at their conclusions, making explain ability a critical component. Hence, it is crucial that models provide clear explanations for their predictions. To address this, we employ frame semantics, a structured approach that extracts key voting-related segments from claims, enabling transparent justifications.

Frame-semantic parsing [5] involves the automatic identification of semantic frames [4] and their corresponding frame elements in text. Semantic frames represent structured events, concepts, or scenarios, each containing frame elements that describe the roles or entities involved. Frame elements are the specific roles or entities that participate in a frame, such as "agent". Lexical units, which uniquely pair words with their meanings, evoke semantic frames. These frames offer a structured approach to natural language processing tasks and have been employed in knowledge extraction [10], question answering [5], and event detection [11].

The project's code repository is at https://github.com/idirlab/claimlens. The data and models needed to reproduce the system can be downloaded from https://drive.google.com/drive/folders/106wSW9y4d1D93wsL7KV7LcDiFHSzG_2H?usp=drive_link. In this work, we establish a foundation for fact-checking political claims related to voting. Our contributions are summarized below.

KDD-UC '24, August 25–29, 2024, Barcelona, Spain

- We developed a system for fact verification using frame-semantics and semantic search, available at https://idir.uta.edu/claimlens/fact-check.
- We used semantic search to find bills for a given claim and performed a detailed analysis on the performance of different semantic similarity models.
- We constructed a novel dataset which maps voting-related fact-checks to their corresponding relevant bills and performed an analysis on several semantic similarity models.

2 DESIGN OF METHODOLOGY

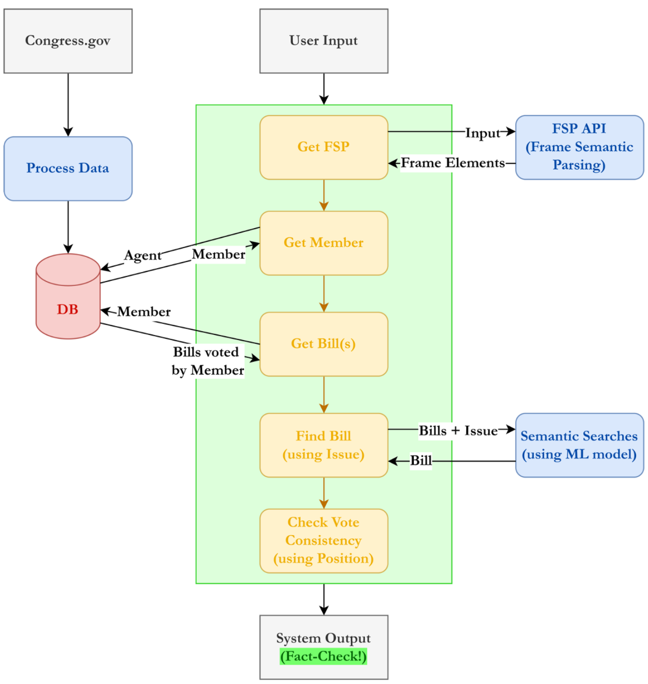

Figure 1: A flowchart describing the overall system.

In light of the critical need for transparency in fact-checking, our approach prioritizes explain-ability as the cornerstone of our system. We achieve this by extracting the frame-semantic structure embedded within each claim. This structural analysis allows our system to dissect a given claim by separating it into its fundamental components, as shown in step one of Figure 1. Specifically, for any given claim, we focus on extracting the voting-related frame-semantic information. In this work, we narrow our focus to three frame elements (FEs): Agent and Issue and the Position FE. The Agent represents the individual whose voting record is in question, while the Issue pertains to the specific bill or legislative matter being voted on. The Position FE represents the stance of the Agent on the Issue (for or against). A more detailed discussion of the Position FE follows in Section 2.3. By leveraging the Agent and Issue FEs, our system can accurately determine whose voting history needs to be investigated and identify the precise issue or bill in question. This comprehensive analysis not only enhances clarity but also strengthens the reliability of our fact-checking process by ensuring

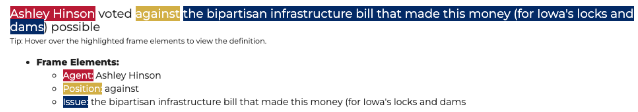

Figure 2: Results of frame-semantic parser on an example claim.

that each claim is thoroughly and transparently evaluated. We show a claim that has been previously fact-checked [1] as an example of the frames in Figure 2.

2.1 Agent Lookup

To verify the voting records of the person mentioned in the claim, it is essential to map the Agent frame element of a claim to a specific congress member as shown as the second step in Figure 1. To do so, we utilize SQL queries to search for congress members whose names match the person mentioned in the Agent FE. The queries are designed to handle different variations of the agent name:

- If only a last name is found, the query searches for database records containing that last name.
- If both a first and last name are identified, the query looks for records containing both.
- If multiple first names are found, the query uses OR conditions to search for records that match any of the first names and an AND condition to match the last name.

The function executes the constructed SQL query on the database and retrieves the member IDs of the matching congress members. This enables the mapping of the Agent frame element to a specific congress member for further verification of their voting records. When two members have similar names (for example if the system finds only the first name John in the claim and there are multiple congress members named John), we select the more recent one based on their records. This process faces several challenges. The first one is the usage of nickname, aliases and preferred or shortened names. Often, in natural language, the full name of people is not used and people are referred by an different name. Similarly, claims often use aliases, such as "Sleepy Joe" for Joseph R. Biden or "Wacky Jacky" for Jacky Rosen. To tackle this, we extracted two lists of commonly used nicknames from Wikipedia. [2] [3] These lists help map nicknames to the actual names of congress members. While not exhaustive, they provide a robust starting point for this mapping. Secondly, many congress members also use shortened names (e.g., Joe instead of Joseph) or preferred names (e.g., Ted Cruz instead of Rafael Edward Cruz). For example, the claim "Bob Casey voted in favor of deadly sanctuary cities that released thousands upon thousands of illegal alien criminals and vicious gang members to prey on Pennsylvania streets, to prey all over this country" refers to the Senator Robert P. Casey, Jr. from Pennsylvania. To tackle this, we use a list of congress members' preferred names and a list of common preferred names [4] for other cases. Finally, another

[1] https://www.politifact.com/factchecks/2022/jan/21/liz-mathis/hinson-joined-letter-iowa-lock-and-dam-money-she-d/
[2] https://en.wikipedia.org/wiki/List_of_nicknames_of_presidents_of_the_United_States
[3] https://en.wikipedia.org/wiki/List_of_nicknames_used_by_Donald_Trump
[4] https://pypi.org/project/nicknames/

Figure 3: Congresswoman Ashley Hinson as found by the agent lookup. The system displays the Congress Member ID along with a picture and other details pulled from Wikipedia.

challenge is name disambiguation from common pronouns (he/she). We choose to use a simplification of not disambiguating common pronouns and hence it is not supported by our system. We prefer to process self-contained claims where the context is sufficient to identify the individual. This decision simplifies the task by assuming that each claim provides enough information to identify the congress member without additional context. Continuing with the example claim mentioned in Figure 2, the Agent Lookup found the correct Congress member being referred in the claim and is shown in Figure 3 along with additional details about the member pulled from Wikipedia.

On the evaluation set described in Section 3.2, our approach correctly identifies the Congress member with an accuracy of 86.07%.

2.2 Semantic Bill Search

The next steps involve identifying the bills that are relevant to the Agent and the Issue extracted, as depicted in the third and fourth steps of Figure1. After extracting the Issue frame elements, we find that they may be broadly classified into three categories — a broad idea (e.g., veteran benefits), an action (e.g., funding abortion) or the affect of a bill (e.g., increase in gas prices). To facilitate the identification of relevant bills, we require a methodology that can effectively match the extracted Issue frame elements with corresponding bill texts. BM25 is a commonly used algorithm in information retrieval based on the frequency of matching words in a query and the corpus — something that aligns with what we are trying to achieve. BM25 has proven to be successful in multiple retrieval based tasks [8, 9]. However, for our task, there are cases where claims do not include the exact phrases as used in the bills. For example, consider the claim shown in Figure 2. One of the relevant bills that can be used to verify it is the Infrastructure Investment and Jobs Act, 2021 [5] but the bill does not include the phrase "made this money (for Iowa's locks and dams) possible" in the text. The language used in writing bills is usually not the same as natural language, as it tends to be formal, precise, objective, technical and often complex. Given this reason, we could not rely solely on keyword search for finding relevant bills. Hence, we use semantic search methods to extract the semantic meaning from the given Issue frame element and the bill corpus to get the bills that most closely align with the given issue semantically. Additionally, to find as many relevant bills as possible, we retrieve the *top k = 10* bills.

2.3 Vote-Claim Alignment

The final step involves determining whether the expressed position on an issue and the topic of the claim (Issue FE) align with the description of the bill and the actual vote cast (Yea or Nay). This process faces two major challenges:

- First, the system cannot rely only on the vote and the Position FE. A claim may not have any Position FE at all in which case the claim supports the issue outlined in the Issue FE (for example, a claim of the format <Agent> voted to <Issue> implies that the intended position of the Agent is in favor of the issue). Moreover, the bills are found solely using the Issue FE and the vote on a bill and the Position FE may not mean the same thing. This entails the fact that voting Yea on a bill may not mean that the individual supports the Issue and vice versa. For example, consider the following claim:

  `<Agent> voted against access to abortions.`

  Say the Agent voted "Yea" on a bill that bans abortions. In this case, the "Yea" vote shows the intent to ban abortions and means the opposite of "access to abortions"(the Issue FE).
- Second, finding if the claim is supported or opposed by the given vote is difficult. It is important to understand whether intended position of the agent (implicitly or explicitly stated) and issue together are consistent with the bill found and the vote on the bill cast by the Agent together. Using the aforementioned example, the Agent is "against access to abotions" and voted "Yea" on a bill that bans abortions. The claim and the Agent's vote are consistent with each other as both ultimately mean that the Agent intends to ban or limit access to abortions.

Given the complex nature of this task, we utilize large language models to perform a classification task as described in Section 4.3.

3 DATASETS

3.1 United States Congress Dataset

To build a dataset for verifying voting-related claims, we needed a database of bills and voting records. We collected and parsed available bills, votes, and congress member information from the official US voting records. [6] This extensive data collection process involved gathering and processing a vast amount of legislative information spanning over two centuries. Our collected voting records encompass a total of 493 congress members, covering a time period from 2013 until 2024. In addition to the congress member data, our dataset includes 63,498 bills from 2013 until 2023. Each bill includes information about the Congress number where it was put forward,

[5] https://www.congress.gov/bill/117th-congress/house-bill/3684

[6] https://www.congress.gov/

Model	Recall @ 10
Dataset Max Baseline	0.5676
msmarco-distilbert-base-tas-b*	**0.1760**
msmarco-MiniLM-L-6-v3△	0.1689
msmarco-roberta-base-v3△	0.1630
msmarco-MiniLM-L-12-v3△	0.1537
msmarco-distilbert-base-v4△	0.1444
msmarco-distilbert-base-v3△	0.1293
msmarco-roberta-base-ance-firstp*	0.1160
msmarco-distilbert-base-dot-prod-v3*	0.1134
BM25Okapi	0.0475
BM25L	0.0380
BM25Plus	0.0348

* Models tuned for dot product
△ Models tuned for cosine similarity

Table 1: Evaluation of different semantic search models. BM25 implementations are included to serve as baselines.

the type of the bill, the bill number and the textual description which is used for semantic search as described in Section 4.2. The types of the bills include HRES, HCONRES, HR, S, HJRES, SCONRES, SJRES and SRES. For example, 118 H.RES.115 refers to the Bill number 115 of type H.RES.— House Resolution—of the 118th U.S. Congress. We also collected and integrated 1,141,918 individual votes cast by congress members on 2716 bills [7] from 2013 until 2022. These votes represent the actual voting records of congress members on specific bills that have been put to a vote.

3.2 Bill Matching Evaluation Set

We collected fact-checks from online sources such as PolitiFact, AP Fact Check, FactCheck.org, etc.. Then, we narrowed them down by choosing sentences that contain variations of the word "vote" (namely votes, voted, voting and vote). Every Politifact fact-check contains a section which highlights a list of the sources used in the verification process. We collected these sources and produce a new evaluation dataset for the bill search model. We extracted the URLs in these sources that map to a US Congress rollcall or bill text for every voting-related claim. The dataset finally allows us to evaluate the bill search system - as discussed in Section 4.2 - as it is a mapping between a voting-related claim and the relevant bills used to verify it.

4 MODELS

4.1 Frame-Semantic Parsing Model

Frame Semantic Parsing plays a pivotal role in achieving the desired level of explain-ability within the fact-checking system. To effectively meet the objectives of our task, it was imperative to employ a frame semantic parsing model that exhibits robustness in identifying the pertinent frame elements while simultaneously demonstrating sufficient performance to satisfy the speed requirements dictated by the system's intended use case. For these reasons, we use the model proposed by [3] given its robust nature and its open-source code.

To evaluate our frame-semantic parsing (FSP) model, we employed a systematic approach using the evaluation set described in Section 3.2. First, we processed these claims through our FSP model to extract the frame elements for each claim. Subsequently, we manually labeled the model's output to assess its exact match accuracy in identifying three key frame elements: agent, position, and issue. This labeling process involved determining whether the model had correctly identified each frame element within the claims. Following this evaluation, we calculated the exact match accuracy for each frame element individually. The results demonstrated exceptional performance across all three categories, with the model achieving 97.46% accuracy for agent identification, 92.4% accuracy for position extraction, and 88.6% accuracy for issue recognition. These findings indicate that our FSP model exhibits robust capabilities in accurately parsing and categorizing the core components of political claims within our evaluation framework.

4.2 Bill Search Model

Given the reasons discussed in Section 2.2, we use Semantic Search to find bills that align most closely with the given issue. We found that the average length of all bill descriptions is 111.258 vs the average length of the queries in the evaluation set is 15.468. To address the same, we use models trained for Asymmetric Semantic Search given that bill description are nearly always longer than the given voting-related claims in our dataset. There are multiple steps involved before we search for bills against a query:

- First, we generate *n-dimensional* (depending on the embedding model) embeddings for all bill descriptions available in the dataset and store them with their respective Bill IDs.
- Second, given an issue and a Congress person, we retrieve all the bills along with their embeddings they have voted on. We do this to narrow the search space down to only the bills relevant to the Congress person being referred to in the claim.
- Next, we generate the embedding for the identified issue from the claim
- Finally, we retrieve the *top k = 10* bills based on one of the two following measures — cosine similarity or dot product.

Models tuned to find the closest matches based on cosine similarity and dot product prefer the retrieval of shorter and longer passages respectively. We test both to compare their performance over the evaluation set. Further, we use BM25 implementations defined in [2, 13] as a keyword search baseline and use Recall at 10 (R@10) as the evaluation metric. The results are shown in Table 1. Using R@10, we found that on average, the models tuned for cosine similarity perform better than the those tuned for dot product. However, the best performing model, for our task is `msmarco-distilbert-base-tas-b` which is tuned for dot product. We also observed that all models used in this approach exceed the BM25 baselines; however, there is room for improvement. An example of one of the retrieved bills for the claim in Figure 2 is shown in Figure 4.

4.3 Claim Alignment Model

After obtaining a list of bills related to a specific issue and the corresponding votes on those bills, the next step is to analyze the

[7] These are the bills which have been voted on.

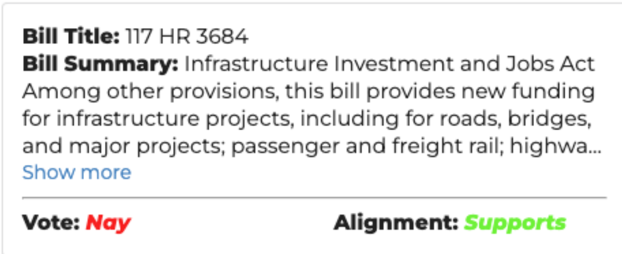

Figure 4: A bill found by the bill search model for the claim shown in Figure 2.

retrieved information to determine if it can be used to substantiate the claim being made. This involves comparing the votes to the stance taken in the claim, as outlined in Section 2.3. By aligning the voting records with the position asserted in the claim, we can assess whether the available data supports or contradicts the claim in question.

In our approach, we harness the advanced natural language processing capabilities of large language models (LLMs) to determine the congruence between a specific bill and a corresponding vote, in relation to the overarching claim being assessed. By leveraging the LLM's ability to comprehend the semantic meaning of the bill's content and the implications of voting in favor of or against it, we can ascertain whether the stance taken by the vote aligns with or contradicts the assertion put forth in the claim. This approach enables a comprehensive evaluation of the claim's truthfulness.

We conducted a qualitative assessment to evaluate the performance and user experience of several Large Language Models (LLMs), including Claude 3 (Opus, Sonnet, and Haiku variants), Llama 3 (70B), GPT-3.5, GPT-4, and GPT-4o. Our analysis, which focused on overall impressions and subjective quality measures, indicated that GPT-4 and Claude 3 Opus frequently aligned with human judgement. In contrast, GPT-4o and Llama 3 exhibited slightly more deviations from anticipated outcomes. Based on the favorable cost-to-performance ratio, we opted to integrate GPT-4o into our system. For the claim in Figure 2 and the vote cast by the Congress member on the retrieved bill in Figure 4, the alignment is shown in Figure 4.

5 CONCLUSION AND FUTURE WORK

In this work, we introduced a system which utilizes frame semantic parsing and semantic search for automated and explainable fact checking. We also highlighted integral challenges and mentioned our approach to solve them, specifically, semantic bill search and vote-claim alignment. We also built a US Congress bills and votes dataset and an annotated bill matching evaluation set.

In the future, we will work on making the bill search model more robust by constructing a dataset bigger than but similar in structure to the evaluation set we currently have and fine tuning multiple models on the new dataset. To enhance the bill search model's comprehension of bill descriptions, we intend to employ a separate model to generate relevant tags based on the content of each bill's description. These tags will be prepended to the corresponding bill descriptions, providing additional context and improving the search model's ability to accurately interpret and retrieve relevant bills.

We also plan to explore alternatives to Large Language Models for the task of vote-claim alignment to address the computational efficiency requirements of our system. Of particular interest is the reformulation of this challenge as a textual entailment problem, which will allow us to leverage the extensive body of research and methodologies developed within the textual entailment domain. By doing so, we aim to identify and implement more computationally efficient approaches that maintain the desired level of accuracy in aligning votes with their corresponding claims, ultimately enhancing the scalability and real-time performance of our system.

Finally, to expand the scope of the system beyond voting-related claims, we plan to build a larger and more diverse dataset that encompasses a wider range of claim types and evidence types. This will involve getting claims from various domains, such as healthcare, economics, social issues, and international affairs. To accommodate this expanded scope, we will explore the utilization of additional frames types that are relevant to the new claim types. By incorporating these additional frames, we can improve the system's ability to understand and represent the nuances of claims across different domains. Furthermore, we will investigate and develop new techniques for finding evidence to support the verification of claims in these expanded domains. This may involve leveraging domain-specific knowledge bases, reputable news sources, etc. to gather relevant and reliable evidence. By increasing the scope of the system to handle claims beyond voting-related issues, building a larger and more diverse dataset, utilizing additional relevant frames, and developing new evidence-finding techniques, we aim to create a more comprehensive and robust fact-checking system.

REFERENCES

[1] Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. Feverous: Fact extraction and verification over unstructured and structured information. *arXiv preprint arXiv:2106.05707* (2021).

[2] Dorian Brown. 2020. *Rank-BM25: A Collection of BM25 Algorithms in Python*. https://doi.org/10.5281/zenodo.4520057

[3] 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), to appear*.

[4] Charles J. Fillmore and Collin Baker. 2009. 313 A Frames Approach to Semantic Analysis. In *The Oxford Handbook of Linguistic Analysis*. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199544004.013.0013 arXiv:https://academic.oup.com/book/0/chapter/335357314/chapter-ag-pdf/50250673/book_38640_section_335357314.ag.pdf

[5] Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics* 28, 3 (2002), 245–288. https://doi.org/10.1162/089120102760275983

[6] Dan S Nielsen and Ryan McConville. 2022. Mumin: A large-scale multilingual multimodal fact-checked misinformation social network dataset. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 3141–3153.

[7] Huyen Trang Phan, Ngoc Thanh Nguyen, and Dosam Hwang. 2023. Fake news detection: A survey of graph neural network methods. *Applied Soft Computing* (2023), 110235.

[8] Ronak Pradeep, Xueguang Ma, Xinyu Zhang, Hang Cui, Ruizhou Xu, Rodrigo Frassetto Nogueira, and Jimmy Lin. 2020. H2oloo at TREC 2020: When all you got is a hammer... Deep Learning, Health Misinformation, and Precision Medicine. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020 (NIST Special Publication, Vol. 1266)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec29/papers/h2oloo.DL.HM.PM.pdf

[9] Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. 2021. Yes, BM25 is a Strong Baseline for Legal Case Retrieval. arXiv:2105.05686 [cs.IR]
[10] Anders Søgaard, Barbara Plank, and Héctor Martínez Alonso. 2015. Using Frame Semantics for Knowledge Extraction from Twitter. In *AAAI Conference on Artificial Intelligence*.
[11] Evangelia Spiliopoulou, Eduard Hovy, and Teruko Mitamura. 2017. Event Detection Using Frame-Semantic Parser. In *Proceedings of the Events and Stories in the News Workshop*. Association for Computational Linguistics, Vancouver, Canada, 15–20. https://doi.org/10.18653/v1/W17-2703
[12] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. *arXiv preprint arXiv:1803.05355* (2018).
[13] Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to BM25 and Language Models Examined. *Proceedings of the 19th Australasian Document Computing Symposium* (2014). https://api.semanticscholar.org/CorpusID:207220720
[14] William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648* (2017).
[15] Xia Zeng, Amani S Abumansour, and Arkaitz Zubiaga. 2021. Automated fact-checking: A survey. *Language and Linguistics Compass* 15, 10 (2021), e12438.

A APPENDIX

A.1 Prompt to LLM

We use the following prompt with the Description, Vote Type and Claim filled in as a prompt to the LLM:

> 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 Description: {Description}
>
> 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.

A.2 Reproducibility

A.2.1 Data. The data and models needed to reproduce the system can be downloaded at https://drive.google.com/drive/folders/106wSW9y4d1D93wsL7KV7LcDiFHSzG_2H?usp=drive_link.

A.2.2 Code.

- Clone the repository at https://github.com/idirlab/claimlens.
- Place "congress.db" downloaded in A.2.1 under claimlens/data.
- Place the frame semantic parsing model downloaded in A.2.1 under claimlens/models.
- Create a virtual Python environment using

  ```
  python3.10 -m venv env
  ```

  This step may be different depending on the OS.
- Activate the virtual environment using

  ```
  source env/bin/activate
  ```

  This step may be different depending on the OS.
- Install Python packages using

  ```
  pip install -r requirements.txt
  ```

- For the API calls to OpenAI, it will be required to follow the instructions at https://platform.openai.com/docs/overview to generate the required API keys and place them in

  ```
  .env
  ```

- Run the command

  ```
  uvicorn main:app -reload
  ```

The final step starts locally running a local FastAPI server to which HTTP requests can be sent to receive system outputs.