ClaimLens: Automated, Explainable Fact-Checking on Voting Claims Using Frame-Semantics

[1]Jacob Devasier [1]Rishabh Mediratta [1]Phuong Anh Le
[2]David Huang [2]Chengkai Li
University of Texas at Arlington
[1]{jacob.devasier, rxm5684, phuonganh.le2}@mavs.uta.edu;
[2]{david.huang, cli}@uta.edu

Abstract

We present ClaimLens, an automated fact-checking system focused on voting-related factual claims. Existing fact-checking solutions often lack transparency, making it difficult for users to trust and understand the reasoning behind the outcomes. In this work, we address the critical need for transparent and explainable automated fact-checking solutions. We propose a novel approach that leverages frame-semantic parsing to provide structured and interpretable fact verification. By focusing on voting-related claims, we can utilize publicly available voting records from official United States congressional sources and the established Vote semantic frame to extract relevant information from claims. Furthermore, we propose novel data augmentation techniques for frame-semantic parsing, a task known to lack robust annotated data, which leads to a +9.5% macro F1 score on frame element identification over our baseline.

1 Introduction

The proliferation of misinformation and disinformation in today's digital landscape has highlighted the urgent need for effective and efficient fact-checking solutions. Manual fact-checking is time consuming and is often too slow to stop the early spread of misinformation. Automated fact-checking methods have emerged as a promising approach to combating the spread of false information. Early approaches formulated queries for databases (Wu et al., 2014) and knowledge graphs (Ciampaglia et al., 2015); however, with the strength of large language models, most of the existing systems rely on machine learning models (Nielsen and McConville, 2022; Wang, 2017; Thorne et al., 2018; Aly et al., 2021) which suffer from a critical limitation: a lack of transparency and explainability. To alleviate this problem, some systems have incorporated an explanation element (Yao et al., 2023) which generates explanations for their predictions. But these post-hoc explanations can result in the model justifying incorrect predictions or hallucinating facts to justify a correct prediction. The opacity of these models can lead to a trust deficit, making it difficult for users–particularly journalists, researchers, and policymakers–to understand the reasoning behind the fact-checking outcomes. This limitation is particularly concerning in high-stakes domains, such as journalism, healthcare, and finance, where the credibility of fact-checking results is paramount.

Recent works towards automating fact-checking are primarily focused on fake news/misinformation detection (Nielsen and McConville, 2022; Wang, 2017) and fact verification (Thorne et al., 2018; Aly et al., 2021). Fake news detection is generally defined as the identification of news containing non-factual statements, often with malicious intention to mislead the public (Zhou and Zafarani, 2020). This is typically done by building models which look at a combination of features such as linguistic cues, user statistics, and news sources, without necessarily determining the truthfulness of the statements. Fact verification is the process of verifying whether a particular claim is true or false given a piece of evidence (Zeng et al., 2021). Fact verification methods assume that the piece of evidence is given. However, this is not always the case for real-world claims which are often self-contained and lack supporting evidence. Thus, to fact-check a claim, it is necessary to couple fact verification with an effective evidence retrieval method.

In this work, we explore the use of frame-semantics (Fillmore and Baker, 2009)—a structured method of extracting important segments from a sentence—in evidence retrieval and fact verification, in order to produce an end-to-end automated, explainable fact-checking system. Frame-semantic parsing (Gildea and Jurafsky, 2002) is the process of automatically identifying semantic frames (frame identification) and frame elements

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing
System Demonstrations, pages 311–319

(argument identification) within text. Semantic frames are structured events, concepts, or scenarios containing frame elements (FEs) which describe different roles or entities related to the frame. Semantic frames provide a structured framework for performing and explaining natural language processing tasks and has been previously used for knowledge extraction (Søgaard et al., 2015), question answering (Gildea and Jurafsky, 2002), and event detection (Spiliopoulou et al., 2017).

This study focuses on voting-related factual claims, as it is a domain where a large amount of structured, trustworthy data are available in the form of voting records. To do this, we utilize the Vote frame defined in Arslan et al. (2020). Given a particular voting-related claim, we subject it to argument identification to extract the Agent, Issue, and Position FEs, which correspond to the voter, what they are voting on, and what their position is, respectively. (An example claim with its FEs can be found in Figure 1.) The truthfulness of the claim can be verified by corroborating or refuting the extracted FEs using a database of public voting records, specifically the United States Congressional voting records in our system.

The database contains a large number of bills and their descriptions, as well as many congress members and their voting records on the bills. The extracted Agent FEs and Issue FEs are matched with the congress members and bills in the database. While finding the corresponding congress member given an Agent FE is straightforward using a simple keyword search, matching an Issue FE with bills is considerably more challenging. In this work, we analyze several text search approaches for matching Issue FEs to their respective bills. To evaluate these search methods, we collected a new dataset (details in Section 3.2) of voting-related claims from PolitiFact fact-checks and their corresponding bills from the content of the fact-checks.

To perform the frame-semantic parsing, we use the system described in Devasier et al. (2024) for frame identification and build on the work in Zheng et al. (2023) for argument identification. To overcome the limited data for the Vote frame in Arslan et al. (2020), we developed two strategies for data augmentation, including *FE interleaving* and *FE permutation* (detailed in Sections 3.3.1 and 3.3.2, respectively). FE interleaving takes two annotated sentences with the same frame and swaps combinations of FEs between the two sentences to create

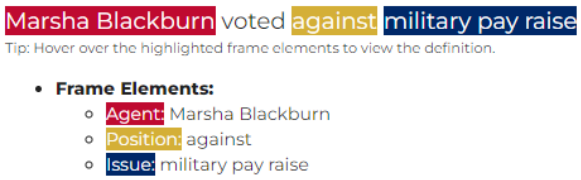

Figure 1: Frame-semantic parse using the Vote frame on a voting-related claim.

new ones. FE permutation uses a single annotated sentence to create new sentences by reordering the FEs in the original sentence.

While voting-related claims is a limited scope, this work can be applied using any frame, given there is sufficient data available, and we plan to expand this work in the future into a few other feasible domains, e.g., verifying claims related to OECD countries using public datasets on their GDPs, crime rates, education rankings, and so on.

We summarize our contributions below.

- We developed the first system for fact verification using frame-semantics, available at https://idir.uta.edu/claimlens/fact-check.
- We proposed novel data augmentation techniques for frame-semantic parsing, a task that has limited available data due to its annotation difficulty, and we provided detailed evaluations on the techniques' utility using the Vote frame.
- We developed a novel dataset which maps voting-related fact-checks to their corresponding bills and performed a detailed analysis on matching extracted voting issues with their respective bills using several semantic similarity models. This dataset and all source code is available at https://github.com/idirlab/claimlens.

2 Methodology

2.1 Agent Lookup

Mapping a claim's Agent FE to a specific congress member is necessary to verify the voting records of the person mentioned in the claim. For this process, we use SQL queries to find congress members who have names similar to each word in the Agent FE. If there is a conflict where two results are found with the same name, we pick the more recent congress member. There are several challenges that appear with this stage of the system. First, claims often use nick names, such as "Sleepy Joe" (used by some to refer to Joe Biden) or "Meatball Ron" (referring to Ron DeSantis by some). To address this, we extracted two lists of commonly used nicknames

of political figures from Wikipedia (Wikipedia contributors, 2024a,b) as mappings for congress members. These lists are not comprehensive, but should be sufficiently robust. Similarly, many congress members use or are referred to by shortened names (Joe instead of Joseph) or different preferred names (Ted Cruz instead of Rafael Edward Cruz). To address this, we utilize the list of congress members' preferred names along with a list of common preferred names for undocumented instances.

2.2 Semantic Bill Search

Finding the bill described by the extracted Issue FE is a difficult task as the Issue FE can be an abstract topic (e.g., "gun control"), a specific action or bill (e.g., "Inflation Reduction Act of 2022"), or the result of a particular bill (e.g., "preventing women from getting abortions"). Furthermore, it is often the case that bills themselves do not mention colloquial terms used to describe such bills, e.g, the bill STOP School Violence Act of 2018 which would expand access to guns in schools. For these reasons, it is important that evidence retrieval cannot rely solely on keyword search. To support these features, we utilize semantic search to match the semantic meaning of Issue FEs with bills.

2.3 Vote-Claim Alignment

Determining whether a claim is refuted or supported by a given evidence is yet another difficult task due to two primary challenges. First, the system cannot simply match the vote and the Position FE since bills may take a positive/negative stance on an issue, e.g., banning/legalizing it. Second, determining whether a claim is supported or refuted by a vote on a bill requires a strong understanding of the bill and its potential implications.

3 Datasets

3.1 United States Congress Dataset

To build our dataset of bills and voting records, we collected and parsed all bills, votes, and congress members from the official US voting records. Our collected voting records include 12,677 congress members from 1789 until 2024, 271,871 bills from 1973 until 2024, and 6,745,285 votes on 7,055 bills from 1990 until 2023. We only retain the last vote cast on each bill to ensure that our records reflect the congress member's final stance on a bill. To enable efficient searching for congress members and votes, we store the voting records locally in an SQLite database.

Dataset	# Train	# Test
Bill Match	0	79
Vote Frame	75	21
Vote GPT Negatives	81	24
Vote FE Permutation	290	73
Vote FE Interleaved	3,154	2,808
Vote FE HC Interleaved	1,697	2,808

Table 1: Statistics of model training/evaluation datasets. HC indicates that all augmentations have a high linguistic acceptability (CoLA score >0.95).

3.2 Bill Matching Evaluation Set

We collected 1,552 fact-checks which mentioned some form of "vote" from PolitiFact. From this set of fact-checks, we manually extracted 193 claims containing the Vote frame. Each PolitiFact fact-check includes a list of sources used in the fact-checking process. We use these sources to construct a new evaluation dataset for the bill matching model by collecting any URLs to a congressional rollcall or bill for each fact-check. This resulting dataset consists of 79 voting-related factual claims and their corresponding bills used to fact-check them, and it enables the evaluation of bill matching systems by mapping factual claims to relevant bills.

3.3 Frame-Semantic Parsing Dataset

Typically, frame-semantic parsing models are trained using the FrameNet (Fillmore and Baker, 2009) dataset; however, since this study is limited to voting claims, we only used the Vote frame samples annotated by Arslan et al. (2020). This dataset is labeled "Vote Frame" in Table 1.

Because the Vote frame dataset has a limited number of samples, we chose to augment the dataset with additional samples to enable more robust model training. We developed two strategies to increase the diversity of training data for identifying frame elements (argument identification) without the need to manually annotate new sentences, as detailed below. Because the Vote dataset had very few negative samples, we used GPT-3.5 to generate additional sentences which contain some form of *vote* without evoking the Vote frame (Vote GPT Negatives in Table 1).

3.3.1 Frame Element Interleaving

Inspired by computer vision techniques, such as CutMix (Yun et al., 2019), and continual learning (Parisi et al., 2019), we interleave sentences which evoke the same frame by creating new data by swapping FEs between them. Since FEs share semantic roles within a sentence, we hypothesize that this interleaving of sentences enables our model to be more robust to sentence context. For example, consider two sentences with Agent A_1 and Issue I_1, and Agent A_2 and Issue I_2, respectively. We create two new sentences with Agent A_1 and Issue I_2, and Agent A_2 and Issue I_1. This means that for any two sentences with n intersecting frame elements, we can create $2^n - 2$ new sentences. Table 1 shows the resulting dataset (Vote FE Interleaved) statistics.

Furthermore, we also experimented with removing low quality sentences which could be produced by simply stitching two sentences together. To do this, we used a RoBERTa-based (Liu et al., 2019) model finetuned on the CoLA dataset (Warstadt et al., 2018) which predicts the linguistic acceptability of a sentence. We used 0.95 as the positive-class threshold to determine high quality sentences. We refer to this subset of samples as Vote FE HC Interleaved in Table 1.

3.3.2 Frame Element Permutation

Our practical evaluations found that our frame-semantic parsing model (Section 4.1) tends to overfit to the order in which frame elements appear in a sentence. For example, the model was unable to correctly identify the Time frame element in the sentence "In 2002, Joe Biden voted for the Iraq War" while it was able to identify it in the sentence "Joe Biden voted for the Iraq War in 2002". To help the model learn different orders of frame elements in a given sentence, we generated additional sentences using every permutation of the frame elements in a given sentence. This means that if a sentence has k frame elements, we generate $2^k - 1$ additional samples. The resulting samples of this augmentation are referred to as Vote FE Permutation in Table 1. To generate these permutations, we prompted GPT-3.5 to rewrite a given sentence while retaining the same meaning and FEs. Detailed results of this process can be found in Table 4.

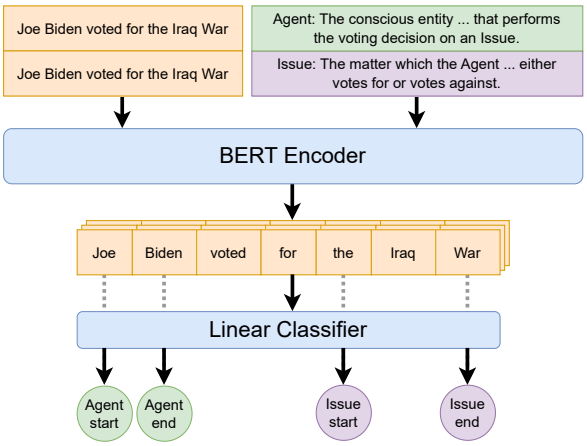

Figure 2: This figure shows the argument identification step of our frame-semantic parsing model. Each frame element is encoded separately with the input sentence and passed to the model. The embeddings are classified into start and end positions for the frame element.

4 Models

4.1 Frame-Semantic Parsing Model

To identify voting-related claims by identifying Vote frames, we utilize the frame-semantic parsing system described in (Devasier et al., 2024). The frame identification component follows a generate-then-filter approach, initially generating candidate targets based on their lemma. A learned classifier then filters these candidates, retaining only those likely to evoke a frame. This two-step method ensures a balance between coverage and precision, first casting a wide net and subsequently refining the selection based on learned patterns.

Our argument (FE) identification model uses an approach similar to AGED (Zheng et al., 2023). AGED defines the FE identification task as a text span identification task wherein a classifier is used to predict the start and end tokens for each FE. Deviating from AGED's approach, we treat each frame-FE pair as a unique input sample, as shown in Figure 2, rather than passing all frame elements in at the same time. This allows the model to individually learn each FE and does not assume that the annotations are complete for all FEs, which may be the case due to the data augmentation process.

4.1.1 Frame Element Partitioning

The output of the FE identification model consists of start and end token probabilities for each frame element. To determine the optimal spans, we evaluate all possible combinations of the predicted FEs. Unlike the greedy algorithm used by AGED, which

Model	Frame Acc	FE Acc	FE $F1_M$
Random baseline	0.488	0.254	0.074
Most frequent baseline	0.974	0.372	0.060
Baseline w/o GPT neg.	0.974	0.853	0.613
Baseline	0.981	0.827	0.537
w/ FE *itl.*	**0.998**	0.851	0.681
w/ HC FE *itl.*	0.993	0.854	0.641
w/ FE *perm.*	0.962	0.845	0.637
w/ FE *itl.* + FE *perm.*	**0.998**	0.875	0.630
w/ HC FE *itl.* + FE *perm.*	0.990	**0.889**	**0.708**

Table 2: Evaluation of frame-semantic parsing models. Frame element interleaving and permutation augmentations are indicated by *itl.* and *perm.*, respectively.

selects spans with the highest scores, we maximize the total prediction score across all spans. Thus, it mitigates the risk of suboptimal selections inherent in the greedy approach.

4.1.2 Ablation Study

We perform an ablation study on our frame-semantic parsing system by training the model with each augmentation for 20 epochs and use the best performing checkpoints for each resulting model. To evaluate the overall performance across the test set we use accuracy for both frame and argument identification. Because of the imbalanced class distribution, we also evaluate the performance for each FE using macro-averaged F1 score. The results of these experiments can be found in Table 2.

First, we found that using GPT negative samples slightly improved the frame identification part of the model, though it led to lower FE accuracy and macro F1 score. Second, We found that each of our augmentation methods increased the macro F1 over both baselines. FE interleaving contributed the most to the performance gain on frame and argument identification, likely due to the volume of data generated (40x the original training set), though there was very little change in FE accuracy. Limiting the FE interleaving to only sentences with high CoLA scores showed less improvement. Only using FE permutation slightly improved the performance on FE macro F1 score. Finally, combining the two strategies improved the system the most, with high-CoLA interleaving performing the best.

4.2 Bill Search Model

As discussed in Section 2.2, we utilize semantic search to find bill descriptions which have the highest semantic similarity. We experimented with models trained on two types of similarity metrics, cosine similarity and dot product.

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-distilbert-base-v4△	0.1444
msmarco-roberta-base-ance-firstp*	0.1160
msmarco-distilbert-base-dot-prod-v3*	0.1134
BM25Okapi	0.0475

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

Table 3: Evaluation of different semantic search models.

To establish a baseline, we also implemented a traditional keyword search model using Okapi BM25, which ranks documents based on term frequency and inverse document frequency, adjusted for document length. We evaluated the models using Recall at 10, a metric that indicates the whether the top 10 results contains the correct bill.

The results, summarized in Table 3, demonstrate that all semantic search approaches outperform the BM25 baseline. Notably, models optimized for cosine similarity generally achieve better performance compared to those optimized for dot product. However, an exception is the DistilBERT-TAS-B model (Hofstätter et al., 2021), which, despite being tuned for dot product, showed the best results.

4.3 Claim Alignment Model

To verify claims by aligning them with relevant legislative votes, we retrieve a list of bills related to a given issue and analyze the associated voting records. Ideally, expert human judgment would be employed for this verification process; however, Large Language Models (LLMs) provide a practical and scalable alternative. In this step, we utilize LLMs to determine the alignment between the content of the bills, the implications of voting for or against them, and the stance of the claim.

The primary function of the LLMs in this context is to parse the language and nuances of the bills and votes, determining whether they support or contradict the given claim. This involves understanding the bill's content, the consequences of different voting outcomes, and the position stated in the claim. Furthermore, our system is designed to generate explanations for each alignment deci-

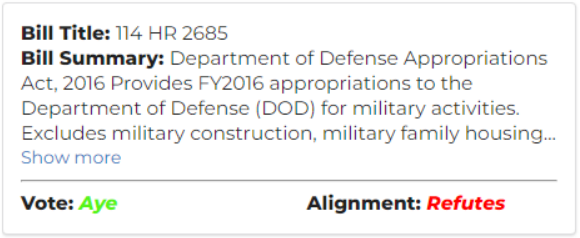

Figure 3: An important bill found by our bill search model on the demo claim. The alignment for this bill is "Refutes" based on the LLM's prediction.

Figure 4: Results of our agent lookup function based on the Agent "Marsha Blackburn".

sion, providing users with transparent reasoning behind the conclusions drawn by the LLMs.

We conducted a qualitative assessment to compare the performance of several LLMs, including Claude 3 (Opus, Sonnet, and Haiku variants), Llama 3 (70B), GPT-3.5, GPT-4, and GPT-4o. The evaluation criteria focused on the models' agreement with human judgment. Our findings indicated that GPT-4 and GPT-4o, along with Claude 3 Opus, consistently demonstrated a higher concordance with human evaluations than the other models tested. Given the comparable performance and a favorable cost-to-performance ratio, we selected GPT-4o for our implementation. We have included the specific prompt used in Appendix A.3.

5 Demonstration

In this section, we demonstrate the functionality of our system using the fact-checked claim, "Marsha Blackburn voted against a military pay raise," as cited in (Greenberg, Jon, 2018). The demonstration showcases the key components of our system, from claim analysis to evidence retrieval and alignment.

First, the system analyzes the semantic structure of the claim to identify the key elements, specifically the Agent (Marsha Blackburn) and the Issue (military pay raise), as illustrated in Figure 1. The Agent lookup process involves retrieving information about the relevant congress member, including their unique identifier, an image, and a brief biography from Wikipedia, as shown in Figure 4.

Next, the system searches for legislative bills related to the identified Issue. It retrieves the voting records of the specified congress member on these bills. For each relevant bill, the system computes the alignment between the claim and the vote, utilizing the methodology discussed in Section 4.3.

Figure 3 shows one of the resulting bills from our bill search model including the bill title/identifier, a summary of the bill, the congress member's vote on the bill, and the alignment of the claim to the bill. In this example, Marsha Blackburn voted for the Department of Defense Appropriations Act of 2016, which specifically includes provisions for military personnel. For this bill, our claim alignment model determined that this vote refutes the claim because "The bill summary indicates that the Department of Defense Appropriations Act, 2016 provides appropriations for Military Personnel, which would generally include funding for military pay raises. Marsha Blackburn's vote was 'Aye', meaning she voted in favor of this bill. Therefore, the claim that 'Marsha Blackburn voted against military pay raise' is incorrect as per this voting record."

6 Conclusion and Future Work

In this work we introduced ClaimLens, the first system which utilizes frame-semantic parsing for explainable, automated fact-checking. Additionally, we outlined important challenges and detailed our methods to solve them, namely on semantic bill search and vote-claim alignment. We also constructed and released our US congress database and our annotated bill matching evaluation set. Furthermore, we introduced and evaluated two novel data augmentation techniques for frame-semantic parsing which significantly improve the model's performance. These achievements lay the foundation for explainable, automated fact-checking with frame-semantics.

In a future study, we aim to expand the scope of the fact-checking capabilities using other frames in (Arslan et al., 2020). One such example is the Occupy_rank frame which is about Items occupying a certain Rank within a hierarchy. For example, consider the claim "The U.S. has the 6th highest poverty rate among OECD countries." Using this frame, we could extract "The U.S." as the Item, "6th" as the Rank, "poverty rate" as the Dimension, and "OECD countries" as the Comparison_set. Then, a query could be formed to deter-

mine whether the claim is true.

We also plan to investigate alternatives to LLMs for vote-claim alignment due to speed demands for our system. Specifically, we would like to represent this as a textual entailment task to utilize the vast research available on textual entailment methods. Finally, we would also like to apply our data augmentation techniques to the original FrameNet dataset to evaluate of the generalizability of our augmentation techniques.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

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.

Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking from knowledge networks. *PloS one*, 10(6):e0128193.

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*.

Charles J. Fillmore and Collin Baker. 2009. 313 A Frames Approach to Semantic Analysis. In *The Oxford Handbook of Linguistic Analysis*. Oxford University Press.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Greenberg, Jon. 2018. Tennessee democrats slam blackburn on military pay vote, overlook her track record of support. https://www.politifact.com/factchecks/2018/jun/05/tennessee-democratic-party/tennessee-democrats-slam-blackburn-military-pay-vo/. [Online; accessed 3-June-2024].

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. *CoRR*, abs/2104.06967.

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.

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*, pages 3141–3153.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

Anders Søgaard, Barbara Plank, and Héctor Martínez Alonso. 2015. Using frame semantics for knowledge extraction from twitter. In *AAAI Conference on Artificial Intelligence*.

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*, pages 15–20, Vancouver, Canada. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. 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.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

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].

You Wu, Pankaj K Agarwal, Chengkai Li, Jun Yang, and Cong Yu. 2014. Toward computational fact-checking. *Proceedings of the VLDB Endowment*, 7(7):589–600.

Barry Menglong Yao, Aditya Shah, Lichao Sun, Jin-Hee Cho, and Lifu Huang. 2023. End-to-end multimodal fact-checking and explanation generation: A challenging dataset and models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2733–2743.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032.

Xia Zeng, Amani S Abumansour, and Arkaitz Zubiaga. 2021. Automated fact-checking: A survey. *Language and Linguistics Compass*, 15(10):e12438.

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 AAAI Conference on Artificial Intelligence*, volume 37, pages 14029–14037.

Xinyi Zhou and Reza Zafarani. 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.*, 53(5).

Limitations

One primary limitation of ClaimLens is its handling of coreference resolution for identifying the Agent Frame Element (FE). The system currently does not support resolving pronouns like "he" or "she," focusing only on self-contained claims that explicitly mention agents. This limitation restricts the system's ability to accurately process claims involving indirect references.

Additionally, while our database includes all roll call votes for each bill, the system only considers the final vote. This simplification may omit important legislative details, such as amendments or preliminary votes, potentially affecting the accuracy of fact-checking. Furthermore, when multiple individuals are associated with the same Agent FE, the system defaults to the most recent congress member, which may not accurately reflect historical actions.

Another limitation of the database is that it requires additional work to maintain up-to-date voting records. While this doesn't cause significant problems to the deployment of the system, additional resources are required to automatically monitor the congressional API for new bills, congress members, and votes, if real-time information is critical.

Finally, the system currently does not incorporate claim metadata, such as the date when the claim was made. This limitation may be impact time-sensitive claims, as the context and accuracy of a claim can change over time.

Frame Element Order	Old Samples	New Samples
Agent, Position, Issue	45	70
Position, Issue, Agent	1	37
Agent, Issue	35	35
Issue, Agent	0	32
Issue, Position, Agent	0	26
Issue, Agent, Position	1	16
Agent, Issue, Position	0	15
Issue, Position, Agent, Time	0	8
Frequency, Agent, Position, Issue	0	7
Time, Agent, Position, Issue	1	7
Agent, Side, Support_rate	4	7
Agent, Position, Issue, Time	4	6
Agent, Position, Issue, Frequency	2	6
Support_rate, Agent, Side	0	6
Agent, Frequency, Position, Issue	2	5
Time, Position, Issue, Agent	0	5
Position, Issue, Frequency, Agent	0	5
Side, Agent, Support_rate	0	5
Position, Issue, Time, Agent	0	5
Frequency, Position, Issue, Agent	0	5
Issue, Agent, Frequency	0	4
Issue, Position, Frequency, Agent	0	4
Position, Issue, Agent, Frequency	0	4
Time, Agent, Issue	2	4
Support_rate, Side, Agent	0	4

Table 4: Detailed statistics of results from FE permutation augmentation.

Ethics Statement

We acknowledge the potential impact of automated fact-checking systems on public discourse and democracy. ClaimLens is designed to be a tool that supports, rather than replaces, human judgment in fact-checking. We encourage users, particularly journalists, researchers, and policymakers, to use the system as a supplementary resource rather than a definitive authority. We are also mindful of the system's limitations and actively work to prevent its misuse, such as the dissemination of misleading information.

A Supplementary Materials

A.1 Detailed UI Information

Figure 5 shows the initial page prompting the user for an input claim to fact-check.

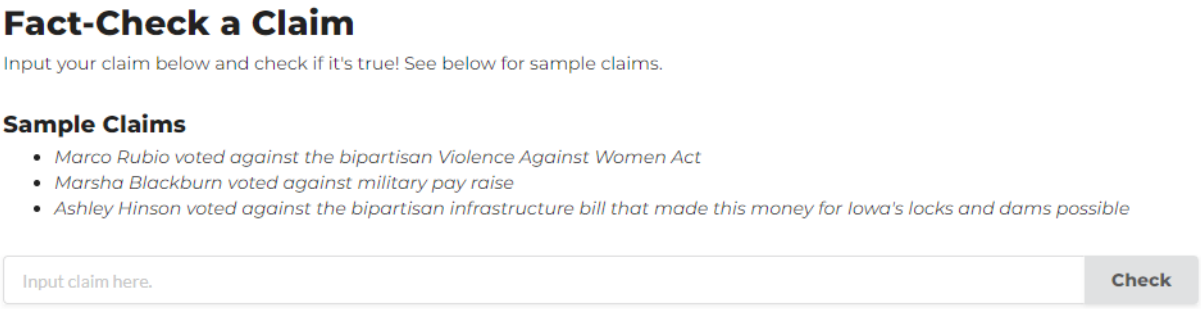

Figure 5: This is the input field to fact-check a claim. Once a claim is entered, the "check" button will run the system on the claim.

A.2 Detailed Augmentation Statistics

Table 4 contains the detailed results of the frame element permutation algorithm in Section 3.3.2.

A.3 Model Prompts

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 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.