Accessible Abstract: Democracy and the free press depends on being able to recognize when facts online are true or not. For machine learning to help this critical problem, it needs good data identifying which statements are backed up by trusted sources and which are not. This research creates a game people can play online to craft difficult claims that can train computers to spot disinformation online.

Links:
- Research Talk [https://youtu.be/JS1lzTsCFTo](https://youtu.be/JS1lzTsCFTo)
- Code and Data [https://github.com/google-research/fool-me-twice](https://github.com/google-research/fool-me-twice)
- Paper Read Aloud [https://youtu.be/zI2O2canp0k](https://youtu.be/zI2O2canp0k)
- Play [http://fm2.qanta.org](http://fm2.qanta.org)

Downloaded from [http://umiacs.umd.edu/~jbg/docs/2021_naacl_fm2.pdf](http://umiacs.umd.edu/~jbg/docs/2021_naacl_fm2.pdf)

Contact Jordan Boyd-Graber (jbg@boydgraber.org) for questions about this paper.
Fool Me Twice: Entailment from Wikipedia Gamification

Julian Martin Eisenschlos, Bhuwan Dhingra, Jannis Bulian, Benjamin Börschinger
Google Research

Jordan Boyd-Graber∗
CS, iSchool, UMIACS, LSC
University of Maryland

Abstract

We release FOOLMETWICE (FM2 for short), a large dataset of challenging entailment pairs collected through a fun multi-player game. Gamification encourages adversarial examples, drastically lowering the number of examples that can be solved using “shortcuts” compared to other entailment datasets. Players are presented with two tasks. The first task asks the player to write a plausible claim based on the evidence from a Wikipedia page. The second one shows two plausible claims written by other players, one of which is false, and the goal is to identify it before the time runs out. Players “pay” to see clues retrieved from the evidence pool: the more evidence the player needs, the harder the claim. Game-play between motivated players leads to diverse strategies for crafting claims, such as temporal inference and diverting to unrelated evidence, and results in higher quality data for the entailment and evidence retrieval tasks. We open source the dataset and game code.

1 Introducing a Game of Challenging Claims

Given a statement—and a large collection of textual knowledge—how do you find evidence that shows a reader that the statement is true or false? This problem takes on multiple forms in the natural language processing (NLP) community. Given only a single statement and a single sentence, this decision process is called recognizing textual entailment (Dagan et al., 2010, RTE) or natural language inference (Bowman et al., 2015; Williams et al., 2018, NLI). Given a single statement and a vast pool of possible evidence (e.g., all of Wikipedia), this problem is called verification (Thorne et al., 2018; Jiang et al., 2020).

∗Work completed while a Visiting Research Scientist at Google.

https://github.com/google-research/fool-me-twice

Figure 1: Overview of the data generation pipeline. In stages 1 to 4, players write challenging claims either entailed or refuted by evidence from Wikipedia (Section 3.1). They are then tasked to spot the refuted claim among a group (Section 3.2). The claims and evidence are available for download.

We review existing resources for the latter task in Section 2 and how they have spawned a vibrant sub-community around related tasks. However, these datasets fail to challenge modern NLP models such as BERT (Devlin et al., 2019) or T5 (Raffel et al., 2020) that achieve “super-human performance” despite also exhibiting “annotation artifacts” that hurt their generalization potential (Gururangan et al., 2018; Tsuchiya, 2018). Our goal is twofold: (1) to build a new, challenging dataset (statistics for FOOLMETWICE in Table 1) that tests models’ abil-
Table 1: Statistics of the FOOLMeTWICE dataset. The train/dev/test split is based on disjoint Wikipedia pages. The number of tokens is an average value computed with a white-space tokenizer. Our dataset is balanced between entailed (true) and not entailed (false) claims.

<table>
<thead>
<tr>
<th></th>
<th>Claims</th>
<th>Entailed Proportion</th>
<th>Pages</th>
<th>Avg # Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>10,419</td>
<td>49.2%</td>
<td>1,811</td>
<td>15</td>
</tr>
<tr>
<td>Dev</td>
<td>1,169</td>
<td>51.0%</td>
<td>209</td>
<td>15</td>
</tr>
<tr>
<td>Test</td>
<td>1,380</td>
<td>49.4%</td>
<td>234</td>
<td>15</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>12,968</td>
<td>49.4%</td>
<td>2,254</td>
<td>15</td>
</tr>
</tbody>
</table>
FEVER has obvious connections to problems in education, journalism, and information science. Thus, it has caught the attention of a subcommunity focused on building systems for FEVER shared tasks. Despite this excitement, Schuster et al. (2019) show that FEVER has many of the same issues as entailment datasets. FEVER has broad or nonsensical claims (Table 2) and many of the claims are generated from the very first line of source Wikipedia documents. This is not just an artifact of crowd-sourcing; a more fundamental problem is that there is no clear definition of what makes a good FEVER example. To date, adversarial FEVER example generation uses automatic rules to increase their difficulty (Thorne et al., 2019). To address these identified weaknesses, Sections 3.1 and 3.2 define a game where the claim writers have a clear objective of “fooling” other human players.

2.2 Gamification for Data Collection

Creating datasets through a fun interactive design is often called gamification. Ipeirotis and Gabrilovich (2014) focus on multiple choice question answering in technical domains such as medicine and rely on redundancy and calibration questions to generate new knowledge. The ESP game (von Ahn and Dabbish, 2004) asks users to write labels for an image that agree as much as possible with other players’ labels.

Another well-known example is protein folding (Cooper et al., 2010), an online game2 that tasks players to twist and bend protein structures, often besting computer algorithms and driving biological innovations (Khatib et al., 2011).

Crucially, these games are either individual or cooperative; in contrast, FOOLMETWICE exploits the adversarial nature of players fooling each other. FOOLMETWICE most closely resembles Balderdash, a board game where players guess which definition of a word is legitimate that is used in information literacy courses (Hays and Hayse, 2017).

In all cases, the intrinsic motivation driven by these games can lead to better outcomes and fewer attempts to “game” the system (Kuznetsova, 2006; Yang and Lai, 2010). Thus our approach constitutes a viable alternative to traditional isolated labelling tasks in crowd-sourcing platforms, where tying payment to completing tasks sometimes hurts final results (Gneezy and Rustichini, 2000).

3 FOOLMETWICE Game Mechanics

This section outlines the two phases of the game: authoring claims (Section 3.1) and voting on those claims (Section 3.2). While these sections present the game in its final form, this is the reflection of an iterative process.

We first began with a paper version (Nielsen, 1989) of the game, which showed that a time constraint made the game feel more fun and encouraged people to not read individual pieces of evidence too intently. Without the timer, people tried to look for tiny clues in text that probably were not there (Wilkinson et al., 2012). We then moved onto a version of the game presented via slides where we experimented with design choices such as the number of claims players distinguish between, and the number of evidence sentences they see while doing that. Examples of the final web interface are shown in Appendix B.

Table 2: Examples from FEVER, which separates entailment examples into three categories. The crowdworkers who authored the examples often edit the first line of the Wikipedia article but not in ways that sound like a plausible hypothesis. We develop a game to build more complex, challenging examples.

Table 3: Top 6 bigrams with the highest LMI (Schuster et al., 2019) for REFUTES in each dataset and each split. Overlapping bigrams are bolded. Compared to FEVER, FOOLMETWICE contains fewer bigrams that “give away” the label on both the train and dev set.
3.1 Crafting Challenging Claims

Our goal is to create a computer game that produces human-authored, interesting, challenging claims paired with evidence that either supports or refutes each claim. One prerequisite for this is that claims avoid high lexical overlap with the knowledge corpus. We thus need to encourage authors to craft claims that cannot be trivially matched to evidence. While this approach has been used for question answering (Wallace et al., 2019; Bartolo et al., 2020), which has a similar retrieval step, to our knowledge it has not been applied to entailment or FEVER.

We recruit users employed at Google, all proficient in English, to play-test the game. At the beginning of each round, we ask each user to generate a true or false statement. We randomly choose a Wikipedia page as a knowledge source and ask them to highlight one or two evidence spans that support (or refute) their claim. They are instructed to write statements that would likely fool other players trying to determine the claim’s veracity quickly and/or without looking at the evidence that support the claim. The reward system defined in the next section is built to be aligned with this objective.

To help authors write hard claims, not entirely similar to the evidence, we show the user what evidence a TF-IDF retrieval system would select from the source and highlight the words that help IR systems select evidence. This implicitly encourages them to craft the claims in a manner such that overlap with the evidence is low (Section 3.2). We include screenshots of the user interface and more details about our design choices in the appendix. Because the players see evidence selected by our retrieval systems, difficult claims for players are also challenging for computers. See Table 3 for a comparison on highly predictive bigrams between FEVER and FOOLME_TWICE (details about how these are computed are in the appendix).

3.2 Spotting the Incorrect Statement

In the game’s second phase, players select the incorrect statement from claims written by other players (Table 4). To separate these two phases of the game, we refer to players in this phase of the game as voters. If a voter can correctly answer quickly (e.g., through their own world knowledge or artifacts), they get up to 120 points, the maximum possible.3

The author and voter split the points: any points the voter leaves “on the table” go to the author. Challenging claims reward the author with more points but easy ones let the voter increase their total.

We do not want to keep claims that are easy to identify as true or false. If the average player can tell through artifacts or common sense that a claim will not be supported, it is uninteresting as an entailment example. For example, if someone sees the claim “Tipper Gore was born in 1048” and remembers that Al Gore was the vice president of the United States in the twentieth century, they can identify that this claim is false. We also want claims that require the voters to carefully read evidence from Wikipedia (Table 4). Voters can ask for hints provided by our evidence selection system (Section 4.1). For each piece of evidence shown, the number of points available to the voters decreases, and points decrease as time progresses as well.

All possible outcomes provide useful information: correct and incorrect choices, with and without evidence. As mentioned before, if voters spot the wrong statement unaided, the claim has underlying issues. When a voter can spot the wrong claim with the help of a particular piece of evidence, then this is a clue that the evidence (and the mechanism that selected it) is useful.

This allows us to specifically optimize for evidence that helps players better answer questions. When voters go from confused to confident about the correct answer, that is a signal that the evidence was effective. When voters select an incorrect answer, that is a signal that the evidence was not effective (or, indeed, misleading).

When voters need more time and evidence and are almost fooled (i.e., nearly think a true statement is incorrect), this is a sign that the statement is challenging for the human–computer team seeking to verify entailment. The statement must be convincingly written, consistent with voter’s world knowledge, and also consistent with the evidence players see. Our game setting helps create conditions where these “tricky” examples can be crafted.

We use two heuristics to ensure quality claims. First, we search for “easy” examples that were consistently solved without inspecting the evidence – however, we were not able to find any. Next, we search for examples which are “too difficult” by computing a maximum a posteriori estimate of the Bernoulli distribution of correct and incorrect votes for each claim. The prior distribution matches the overall accuracy of the dataset (80% of votes are

3Each voting task should take at maximum two minutes, and each point corresponds to a second.
correct) and is equivalent to adding five pseudo-counts (one wrong, four correct) for each question. We use this smoothed estimate rather than the maximum likelihood estimate to account for claims lacking votes. The expected value of that posterior given a Beta(4, 1) prior is (Liu et al., 2012):
\[
\alpha \sim \text{Beta}(4, 1)
\]
\[
\alpha | C_i \sim \text{Beta} \left( 4 + \sum C_i, 1 + \sum (1 - C_i) \right),
\]
where \( i \) sums over the votes, and \( C_i \) is one if the vote was correct and zero otherwise. We analyze all twenty-five claims below a 0.5 threshold and identified three incorrect examples which we subsequently removed.

3.3 Incentive Structure

Players earn points in two ways: either spotting incorrect claims by voting as early as possible or authoring challenging claims. They alternate between the two roles in every game session. These two rewards are in opposition to each other.

Because the goal of the voters is to find the claim that is incorrect, claim authors (of either entailed or refuted claims) only get points when voters are not fooled and when the voters need evidence. The total points are split between the voter and authors when the voter correctly guesses, making this a zero-sum game. As a voter requests evidence or takes more time, a larger fraction of the total points will go to authors. Thus, authors are encouraged to write difficult claims; voters are encouraged to select claims correctly.

When a voter guesses incorrectly, they get no points, to ensure the examples are valid. While incorrect guesses can happen for impossible claims, writing claims that are merely difficult is a better strategy since easy claims that may be spotted quickly are awarded no points.\(^4\)

In addition to humans voting on claims, we also ask users which of the two claims they “like” more, independent of voters’ accuracy. People like true claims (0.39) more than false claims (0.35, \( t = 2.53, p = 0.01 \)), except for claims about science and technology, where people prefer false claims (0.46) more than true claims (0.32, \( t = -2.50, p = 0.02 \)). Authors get points when voters like their claims; this additional incentive encourages authors to create interesting and surprising examples.

4 Methods: Subtasks and Models

Each of the instances in FOOL.METWICE is a tuple \((c, e, l)\): a natural language claim \( c \), evidence \( e \) from a knowledge corpus \( K \) (in our case Wikipedia), and a binary label \( l \) (entailment / contradiction).\(^5\) From this we define two sub-tasks, following Thorne et al. (2018). The first sub-task, retrieval, requires systems to select candidate evidence from \( K \) (including, perhaps, the gold evidence \( e \)). The second sub-task is entailment, where systems given claim \( c \) and the gold evidence \( e \) need to make a final prediction for the label \( l \). We also consider an end-to-end setting. Instead of the gold evidence, systems only have access to the retrieved evidence \( \hat{e} \) at test time. In the rest of this section we define baseline models for each of the sub-tasks.

4.1 Retrieval

Our setting resembles the retrieval setting in the KILT benchmark (Petroni et al., 2021), but the results are evaluated at the evidence level as opposed to the page level, to represent a more realistic use case. The evidence corpus can be found online\(^6\) and consists of twenty-two million text passages, each having a length of a hundred words, from five million pages of the English Wikipedia image from August 2019. We align gold FOOL.METWICE evidence to this knowledge source by selecting the passage with highest overlap with each evidence sentence, according to the modified \( n \)-gram precision component of the BLEU (Papineni et al., 2002). We remove 1598 examples\(^7\) where the precision was less than 0.5.

We evaluate two baselines. The first one follows Chen et al. (2017) and uses a TF-IDF retrieval model with unigrams and bigrams and 2\(^20\) hash buckets. The title of page is added to the passage content for additional context. The second baseline uses Dense Passage Retrieval (Karpukhin et al., 2020, DPR), using the same fixed pre-trained passage embeddings and query encoder as the ones used in Petroni et al. (2021).

4.2 Entailment

For the second component of the task, we follow state-of-the-art entailment models (Zhou et al., 2019; Liu et al., 2020; Eisenschlos et al., 2020):

\(^4\)We also allow players to flag obscene, incorrect, or otherwise problematic claims.

\(^5\)Unlike FEVER, we do not allow authors to write claims that lack “enough information”.

\(^6\)http://github.com/facebookresearch/KILT/

\(^7\)This happens because FOOL.METWICE was constructed from a more recent version of Wikipedia than KILT.
After ending its personal union with Denmark, Iceland was last invaded by the United Kingdom in Operation Fork.

After Elizabeth II’s accession, she changed her house’s name from “House of Saxe-Coburg and Gotha” to the “House of Windsor”, rejecting the “House of Edinburgh”.

Table 4: Claims and evidence shown to players in the voting phase: the voter must detect which claim is incorrect. Initially, the player only sees the claim—if the player can answer with only that, they get the most points. Automatically found evidence is shown one by one upon the voter’s request. Waiting and asking for evidence both decrease the time—and the points—available. Eventually, if time does not run out, the gold evidence selected by the author of the claim is shown.

Table 5: Results of evidence retrieval baselines on FEVER and FOOLMETWICE. \(R\)-Precision is defined as the precision@\(k\), where \(k\) is the number of gold evidence snippets for the claim. FOOLMETWICE is harder for both the retriever systems.

5 Experiment Results: Machines Spotting False Claims

This section studies the performance of existing automatic methods on FM2 for both the retrieval of evidence (Section 5.1) and for entailment once the results are retrieved (Section 5.2).

5.1 Retrieval Results

Retrieving evidence for FOOLMETWICE is considerably harder (Table 5); we also include comparable results on FEVER. The documents retrieved by DPR are consistently better than the ones by a TF-IDF system for both of the datasets we tested, which is consistent with other work on dense text retrieval (Guu et al., 2020).

5.2 Entailment Results

This section presents the results of training a BERT (Devlin et al., 2019) model for the entailment task of FOOLMETWICE. Given a claim and the gold evidence, does the evidence support or refute the claim? To compare with FEVER, we discard all not enough evidence examples, because the lack of evidence for this class makes it trivial to classify correctly.

Following Gururangan et al. (2018), we first train a claim-only classifier, which ignores the evidence text. FOOLMETWICE examples are harder to classify without looking at the evidence (Table 6), indicating that the claims contain fewer...
Table 6: Comparison of dev accuracy between FEVER and FOOLMETWICE for different partitions of the data and when using only claims. The partition into EASY and HARD splits is based on the claim-only classifier: the claim-only classifier can solve EASY examples. FOOLMETWICE examples thus are comparable to HARD FEVER’s difficulty.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Claim-Only</th>
<th>EASY</th>
<th>HARD</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOOLMETWICE</td>
<td>61.9</td>
<td>86.1</td>
<td>66.4</td>
<td>78.1</td>
</tr>
<tr>
<td>FEVER</td>
<td>79.1</td>
<td>97.1</td>
<td>79.3</td>
<td>93.3</td>
</tr>
</tbody>
</table>

Table 7: End-to-end label accuracy results of retrieval followed by entailment on FOOLMETWICE. We vary the number of retrieved examples at prediction time. We compare against using the gold evidence as an oracle, which differs from Table 5 in using a single 100 word passage as evidence.

<table>
<thead>
<tr>
<th>Retriever</th>
<th>Top-1</th>
<th>Top-3</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>69.3</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>62.3</td>
<td>62.0</td>
<td>61.2</td>
</tr>
<tr>
<td>DPR</td>
<td><strong>64.2</strong></td>
<td>63.6</td>
<td>63.9</td>
</tr>
</tbody>
</table>

“give away” artifacts compared to FEVER as already suggested by Table 3. We provide additional discussion in Appendix C.

Like the techniques proposed by Clark et al. (2019), the claim-only classifier can also be used on both FOOLMETWICE and FEVER to split the dev sets into “easy” and “hard” partitions: The EASY partition contains all examples correctly classified by a claim-only classifier, and the HARD partition has everything else. The similar accuracy of the FOOLMETWICE dev and HARD FEVER dev partitions further suggests that FOOLMETWICE is comparable to the harder and higher-quality subset of FEVER (Table 6).

We also train an end-to-end verification model that, rather than taking evidence as given, must use noisy passages from a retrieval system (Section 4.1). At train time, we generate multiple training instances for each claim using either the gold evidence or the top two retrieved examples. At prediction time, we average the logit scores of each of the top-<i>k</i> retrieved passages (Table 7). We include a so-called oracle setting for a fair comparison of the improvement margin. This number differs from Table 6 in that it uses a single gold 100 word passage as evidence instead of short sentences, which might introduce noise.

6 Dataset Analysis: Humans Spotting and Writing False Claims

While the previous section focuses on how well automatic methods can detect false claims, this section focuses on human ability. Voters are usually right and were fooled 20.40% of the time. This section addresses how players are fooled and how this compares to computers.

To provide a better picture of the strategies players use to craft challenging claims, we manually sample fifty instances from the development set that both models and humans answer incorrectly. We focus on these examples because they are the most difficult and are the emphasis of our adversarial technique. Two claims were mislabeled and two more lacked a necessary evidence span. Table 8 shows examples of each of the strategies, which we discuss in more detail in this section.

**Temporal** Many of the most challenging claims require an inference about time: whether one event happened before another, how long an event happened, or whether an event happened during a period. While many of these are based on years, centuries, or other explicit markers of time, some authors use narrative time. For example, the page for the novel *As I Lay Dying* describes the plot in order, so it’s difficult for either a system or a human given sentences (without knowing where they appear in the original page) to know when Addie Bundren dies. This shows some of the limitations of the setup: not only must voters reason across multiple pieces of evidence, this reasoning is only possible if they know the order of the underlying evidence. Other markers of time include “the pilot” for the first episode of The Office; readers must realize that if Kelly Kapoor was introduced in the episode Diversity Day, that implies Mindy Kaling’s character did not appear in the pilot.

**Reasoning** A related, but more general, strategy requires the reader to reason: mathematically, applying definitions, or understanding hyponymy. For example, knowing that the child of your cousin is your second cousin or recognizing that “This mirrors the Disney Parks East regional division consisting of Shanghai Disney Resort, Hong Kong Disneyland and Walt Disney Attractions Japan…” implies that there are more than two Walt Disney resorts outside of the United States.
Paraphrase A well-known strategy to confuse entailment systems is to change words so that there are fewer exact matches. Some of these are straightforward: “Titration is used when doctors test how much sugar is in a patient’s liquid waste” is almost a direct paraphrase of “glucose in urine may indicate diabetes in a patient”. Other paraphrases are more poetic: “Charles Evans Hughes shuffled off this mortal coil in Massachusetts, and then was taken to New York to be submerged in soil” paraphrasing “Hughes died in what is now the Tiffany Cottage of the Wianno Club in Osterville, Massachusetts. He is interred at Woodlawn Cemetery in the Bronx, New York City”. These paraphrases are realistic, similar to how humans might restate facts to make them more accessible or more interesting to a reader.

Controversy A more fundamental issue with entailment systems is that even trusted sources such as Wikipedia contain contradictory evidence. This is most prominent with interpretations of works of fiction, where there are multiple theories about the same work. A skillfully written claim can retrieve one viewpoint while using an opposing viewpoint as the gold evidence.

For example, one claim strongly took the position that the end of the film Inception was a dream. Voters saw evidence to the contrary and thought the claim was refuted. Because systems focus on the highest scoring retrieved passages (as do the human voters), this lead both humans and computers to overlook the disputed interpretations.

6.1 What was difficult for humans?

The amount of evidence a human needs is a unique metric of how difficult a claim is for humans (although incremental evidence is recommended for question answering systems in Boyd-Graber and Börschinger (2020), to the best of our knowledge it has not been applied to entailment or validation). The claims that most challenge humans typically use diversion (e.g., “The Quiet Man was a song by Bing Crosby about a soldier who lost his voice from a bomb in World War 2”), which is particularly challenging for retrieval systems. Other com-
mon strategies for the claims most challenging for humans were paraphrase, which can “hide” the relevant evidence and prevent retrieval, and reasoning, which often requires multiple pieces of evidence to reach a conclusion.

7 Limitations and Conclusion

While this paper seeks to advance the ability of humans and computers to support or refute statements entailed from a static, reliable source, the goal of examining arbitrary statements remains elusive. By construction, we have focused on statements that are incorrect because of factual errors. Other datasets that use human-sourced obfuscations or deception are more nuanced and use framing or shading (Pan and Kosicki, 1993), which models trained on this dataset cannot detect. Our goal is to focus on clear facts that can be recognized by computers, which is already challenging enough.

Further improving verification likely requires creating targeted datasets that focus on specific strategies for creating statements that are refuted by evidence, perhaps selecting different explanations for particular users (Feng and Boyd-Graber, 2019). Likewise, a more complicated task likely requires more nuanced incentives and instructions for authors. However, this dataset provides a foundation to build these richer, more challenging datasets for entailment.

Ethical Considerations

As our work involves human participants, all players provided informed consent and no personally identifiable information (PII) was collected or will be released. The collected data have been vetted for presence of PII as well as offensive language through heuristics and random sampling.

Some participants received fair compensation in the United States in exchange for playing the game, but that compensation was not tied to speed or accuracy to prevent distorting the motivation of players. Intrinsic motivation, such as curiosity, competitiveness, creative drive and fun, rather than extrinsic motivation has been shown to produce higher quality results (Gneezy and Rustichini, 2000).

The released data and the experiments we conducted are in English, therefore we do not claim generalization of our findings across languages. However, we believe that the proposed methods could be applied in other languages using other available corpora as a source of evidence.

Acknowledgements

First and foremost, we would like to specially thank Connie Tao for her guidance and assistance in managing the project. The project would also have been impossible without the FM2 players. We also would want to thank Thomas Müller, William Cohen, Dipanjan Das, Slav Petrov, Pedro Rodríguez, Massimiliano Ciaramita, and Christian Buck for comments on the drafts and testing the game. We also thank the anonymous reviewers for their time, constructive feedback, useful comments and suggestions about this work. Boyd-Graber is supported by NSF Grant IIS-1822494.

References


Appendix

A Experimental Setup

In this section we provide details on the hyperparameters used and the experimental setup. All BERT models described are of base size (12 layers, 16 attention heads, 768 hidden dimension), and contain 110 million parameters.

The training is done for 10 epochs, a learning rate of $10^{-5}$. We use a batch size of 32 and a learning rate of 512. On a single Cloud TPU v2 the model can process one batch 180ms, and a full epoch in around one minute. For all the reported results we take the median over 3 random seeds.

B Game Interface

In this section we include screenshots of the three main screens of the game. Figure 2 shows menu interface that allows players to choose topics according to their interests, we include many Wikipedia categories to ensure a diverse set of options. Figure 3 has example of the voting game, the simplest and fastest way to engage with the game and understand how to be a good author as well. Finally, figure 4 shows the authoring user interface, that displays the retrieved and selected gold evidence as the user types. Matching tokens in the text and the retrieved evidence are highlighted.

C Local Mutual Information

Tables 9, 10 list the top-10 predictive bigrams for the REFUTES label using Schuster et al. (2019)’s method of computing Local Mutual Information (LMI), defined for a bigram $b$ and label $l$ as:

$$\text{LMI}(b, l) = p(b, l) \cdot \log \left( \frac{p(l | b)}{p(l)} \right)$$

where the probabilities use the empirical counts.

Consistent with the much lower claim-only classifier (see Table 6), FOOLMETWICE contains no “give away” bigrams that are highly predictive of the label on both the training and development data whereas, as previously reported by Schuster et al. (2019), FEVER has many. Moreover, the “quality” of predictive bigrams for FEVER suggests that annotators (subconsciously) used specific strategies when writing REFUTES examples (“is only”, “did not”, “is incapable”), but no such patterns can be seen for FOOLMETWICE.

<table>
<thead>
<tr>
<th>Bigram</th>
<th>Train $\text{LMI} \times 10^{-5}$</th>
<th>Dev $\text{LMI} \times 10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>is only</td>
<td>622</td>
<td>938</td>
</tr>
<tr>
<td>did not</td>
<td>859</td>
<td>528</td>
</tr>
<tr>
<td>not a</td>
<td>775</td>
<td>481</td>
</tr>
<tr>
<td>was not</td>
<td>729</td>
<td>–</td>
</tr>
<tr>
<td>incapable of</td>
<td>721</td>
<td>710</td>
</tr>
<tr>
<td>only a</td>
<td>455</td>
<td>717</td>
</tr>
<tr>
<td>is incapable</td>
<td>474</td>
<td>551</td>
</tr>
<tr>
<td>was only</td>
<td>–</td>
<td>536</td>
</tr>
<tr>
<td>has only</td>
<td>447</td>
<td>–</td>
</tr>
<tr>
<td>yet to</td>
<td>420</td>
<td>384</td>
</tr>
<tr>
<td>of being</td>
<td>–</td>
<td>385</td>
</tr>
</tbody>
</table>

Table 9: Top-10 highest LMI bigrams for REFUTES label in FEVER for both Train and Dev. Note the large overlap of label-predictive bigram artefacts.

<table>
<thead>
<tr>
<th>Bigram</th>
<th>Train $\text{LMI} \times 10^{-5}$</th>
<th>Dev $\text{LMI} \times 10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>by a</td>
<td>–</td>
<td>562</td>
</tr>
<tr>
<td>mad ,</td>
<td>–</td>
<td>502</td>
</tr>
<tr>
<td>, mad</td>
<td>–</td>
<td>502</td>
</tr>
<tr>
<td>on the</td>
<td>–</td>
<td>473</td>
</tr>
<tr>
<td>innocent iii</td>
<td>–</td>
<td>467</td>
</tr>
<tr>
<td>statue of</td>
<td>–</td>
<td>426</td>
</tr>
<tr>
<td>for his</td>
<td>–</td>
<td>407</td>
</tr>
<tr>
<td>pope innocent</td>
<td>–</td>
<td>407</td>
</tr>
<tr>
<td>mary ,</td>
<td>–</td>
<td>365</td>
</tr>
<tr>
<td>queen of</td>
<td>–</td>
<td>365</td>
</tr>
<tr>
<td>the second</td>
<td>338</td>
<td>–</td>
</tr>
<tr>
<td>is a</td>
<td>312</td>
<td>–</td>
</tr>
<tr>
<td>was a</td>
<td>307</td>
<td>–</td>
</tr>
<tr>
<td>was the</td>
<td>306</td>
<td>–</td>
</tr>
<tr>
<td>is the</td>
<td>233</td>
<td>–</td>
</tr>
<tr>
<td>of his</td>
<td>200</td>
<td>–</td>
</tr>
<tr>
<td>has never</td>
<td>189</td>
<td>–</td>
</tr>
<tr>
<td>was born</td>
<td>177</td>
<td>–</td>
</tr>
<tr>
<td>written by</td>
<td>165</td>
<td>–</td>
</tr>
<tr>
<td>about a</td>
<td>162</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 10: Top-10 LMI bigrams for REFUTES label in FOOLMETWICE for both Train and Dev. Note both the absence of “give-away” bigram overlap between Train and Dev; and the more “random” quality of predictive bigrams compared to those for FEVER in Table 9.
Figure 2: Menu where players select between authoring or voting on claims. A diverse set of categories is presented to engage people according to their interests.

Figure 3: The voting interface shows one entailed and refuted claim. The player has two decide which one is the refuted one before time runs out. Getting clues consumes 30 seconds in the timer.
Figure 4: In the writing screen, players are asked to write either an entailed or refuted evidence given the evidence on the right hand side. As they write, a retrieval system picks the most relevant evidence. They can mark the gold evidence that supports or contradicts the claim, and are instructed to write in such a way that the gold evidence is not at the top of the retrieved list.