Team Papelo at FEVEROUS: Multi-hop Evidence Pursuit

Christopher Malon
NEC Laboratories America
Princeton, NJ 08540
malon@nec-labs.com

Abstract

We develop a system for the FEVEROUS fact extraction and verification task that ranks an initial set of potential evidence and then pursues missing evidence in subsequent hops by trying to generate it, with a "next hop prediction module" whose output is matched against page elements in a predicted article. Seeking evidence with the next hop prediction module continues to improve FEVEROUS score for up to seven hops. Label classification is trained on possibly incomplete extracted evidence chains, utilizing hints that facilitate numerical comparison. The system achieves .281 FEVEROUS score and .658 label accuracy on the development set, and finishes in second place with .259 FEVEROUS score and .576 label accuracy on the test set.

1 Introduction

The 2021 FEVEROUS (Fact Extraction and VERification Over Unstructured and Structured Information) task (Aly et al., 2021) introduces several challenges not seen in the 2018 FEVER task (Thorne et al., 2018). Tabular information, lists, and captions now appear as evidence, in addition to natural text sentences. Most claims now require multiple pieces of supporting evidence to support or refute them. Even claims that cannot be fully verified now require the submission of supporting evidence for aspects of the claim that can be verified. Counting and numerical reasoning skills are needed to verify many claims.

Annotators for FEVEROUS differed in their interpretation of what constituted necessary evidence, and often added duplicate evidence that should be in an alternative reasoning chain to a main reasoning chain. For this reason it is dangerous to target a precise, minimal set of evidence as in FEVER for high evidence F1 (Malon, 2018), and we instead fill the full set of five sentences and 25 table cells permitted for submission.

Thus we focus on solving the evidence retrieval problem and first assemble a set of preliminary set of relevant facts. Several of these facts may be combined to determine the veracity of the claim. Yang et al. (2018) define multi-hop reasoning as reasoning with information taken from more than one document to arrive at an answer, so using the preliminary evidence set could already be multi-hop reasoning, but from the perspective of retrieval we consider retrieving the initial evidence set to be a first "hop." Where multi-hop reasoning is required, it may be necessary to retrieve additional documents after reading the preliminary evidence, which could not be searched for using the claim alone. We support this functionality by predicting whether evidence chains are complete and generating additional search queries based on the preliminary evidence. This next hop prediction module can be applied as many as seven times to update the evidence chains, each time improving the FEVEROUS score.

On the final evidence chains, the label ("supports", "refutes", or "not enough information") is predicted by a module trained on extracted evidence chains. Because "not enough information" (NEI) labels are scarce, we alternatively can decide whether to give an NEI label based on whether the next hop prediction module is still seeking more evidence for the claim. Inputs are carefully represented to facilitate numerical comparisons for the final label decision and to allow the use of other contextual information by every module. The described system attains a FEVEROUS score of .281 on the development set with label accuracy of .658.

2 Context and structured information

Downstream classifiers usually classify page elements in isolation, but the meaning of these elements sometimes is not clear without contextual information. In the FEVER task, attaching a prefix to each sentence consisting of the page title

Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER) at EMNLP 2021, pages 40–49

Type	Example
Sentence	[Mississippi River] When measured from its traditional source at Lake Itasca, the Mississippi has a length of 2,320 miles (3,730 km).
List item	[Temple Tower] LIST CONTEXT Cast VALUE Marceline Day as Patricia Verney
Table cell	[Temple Tower] VALUE Release date {{ KEY Temple Tower VALUE April 13, 1930 }}
Table cell	[L-arabinose operon] CAPTION Catabolism of arabinose in E. coli {{ KEY Substrate VALUE L-arabinose }} KEY Enzyme(s) VALUE AraA KEY Function VALUE Isomerase KEY Reversible VALUE Yes KEY Product VALUE L-ribulose

Table 1: Example representations of various page elements.

in brackets improved performance (Malon, 2018), for example by providing hints about what pronouns might refer to. We continue this practice for FEVEROUS.

For list elements, we take the page element immediately preceding the list as context. This often is a sentence indicating what is in the list. Then the list element is represented by "[*title*] CONTEXT *context* VALUE *list item*", so that the list element and what the list is about may be seen simultaneously.

For table cells, we represent the entire row containing the cell. If a cell in a row above has an `is_header` attribute, the cells are prefixed with "KEY *header*". This is followed by the actual value from the current row, in the form "VALUE *header*". Thus each cell in a row looks like a combination of key/value pairs (or simply values if there is no header). This representation is similar to the one used by Schlichtkrull et al. (2020). All the cells in a row would look alike if we simply followed this procedure, so we distinguish the key/value pair corresponding to the current cell by enclosing it in double braces. Finally, the title is prepended, and if there is a caption, it is prepended as "CAPTION *caption*". Examples of the table cell, list element, and sentence formats are shown in Table 1.

3 Preliminary evidence retrieval

We follow the baseline system (Aly et al., 2021) to select an initial set of documents for downstream analysis. This module retrieves documents whose titles match named entities that appear in the claim, plus documents with a high TF-IDF score against the claim, up to five total documents.

Following Thorne and Vlachos (2021), we also considered the use of GENRE (Cao et al., 2021) to identify more Wikipedia page titles from entities that were not quite exact matches. (We preferred an exact match if present.) The use of these entities actually drove FEVEROUS score down, perhaps by crowding out the TF-IDF documents, so we reverted to the baseline approach.

Given a set of documents, we rank page elements using models trained to predict the set of evidence elements. One model is trained on sentences, list elements, and table captions, and the other is trained on table cells. We use a RoBERTa base model (Liu et al., 2019) and follow a training approach similar to the Dense Passage Retriever (Karpukhin et al., 2020). Given a positive training pair consisting of a claim c and a piece of evidence e, we collect six negative pairs (c, x_i). For four of the negatives we take x_i to be the highest TF-IDF matches returned by the baseline system that are not part of the gold evidence. For the other two negatives we take x_i to be part of the gold evidence for a different claim, randomly chosen. The multiple choice classification head of RoBERTa outputs a scalar $f(c, x)$ for each pair, and the batch of seven pairs is trained as one example with the cross-entropy loss

$$-\log \frac{e^{f(c,e)}}{e^{f(c,e)} + \sum_{i=1}^{6} e^{f(c,x_i)}} \quad (1)$$

just as in the Dense Passage Retriever. At test time, we run the model on examples of a single claim/evidence pair and collect the scalar $f(c, x)$. These outputs are ranked across all potential evidence to collect five sentences and 25 table cells. Every sentence in the retrieved documents is ranked, but only the top three tables retrieved by the baseline TF-IDF ranker are considered for extracting table cells.

The baseline system extracts sentences and other non-cell elements by TF-IDF similarity to the claim, and table cells with a RoBERTa base sized model that performs sequence tagging on linearized tables. Table 2 compares the recall of our system

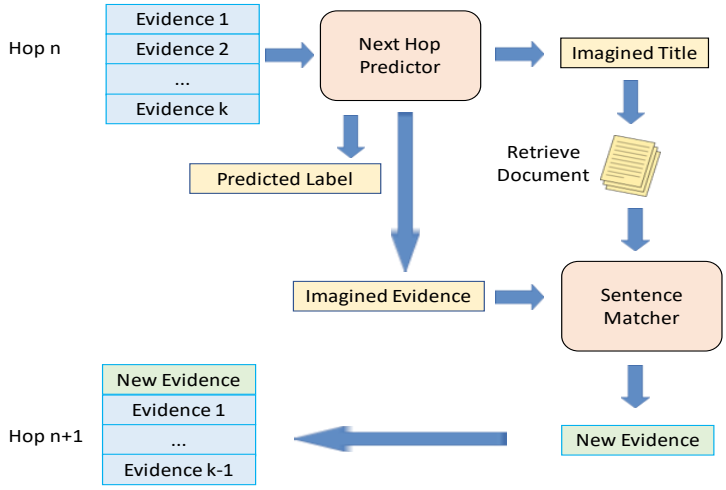

Figure 1: Applying the next hop prediction module to update evidence.

System	Recall
Baseline sentences	.5265
Ranking sentences	.3875
Baseline cells	.2741
Random cells	.2808
Ranking cells	.5028

Table 2: Page element recall.

(top 25 cells and five non-cell page elements) to these modules. This is computed by taking the union of all page elements (cells or non-cells) in all evidence chains in all claims, and considering the fraction that belong to one of our predicted evidence sets for the corresponding claims. We recall more relevant table cells, but surprisingly, fewer relevant sentences. In development, we mistakenly benchmarked the ranking models on a set which made gold evidence available for ranking even if it was not in a retrieved document, and on this basis, it appeared that ranking the sentences was advantageous. Therefore we used not only the table cell ranking module but also the sentence ranking module in our submitted system.

4 Next hop prediction

The use of the evidence ranking model is not sufficient to solve problems that require more difficult kinds of multi-hop reasoning. Though evidence chains are typically rooted in entities and concepts that appear in the claim, as one progresses down the chain it may be necessary to retrieve information about an entity mentioned in a previous piece of evidence. Such information would be difficult to query based on the claim alone.

To support this scenario, we introduce a next hop prediction module, as shown in Figure 1. Hop 1 consists of the evidence retrieved by the evidence ranking module. Given an evidence set produced in hop n, the next hop prediction module attempts to imagine information that is still needed but not retrieved yet. It generates a string consisting of the title of the needed article and the sentence or table cell (in the same format as before) that it wants to retrieve from that article. If available, the article with that title is retrieved; otherwise, sentences from previously retrieved articles will be searched. Then we choose one sentence and two table cells with the best word overlap against the imagined evidence. The bottom ranked elements of the evidence set for hop n are pushed out, and these chosen elements are pushed to the top of the evidence set for hop $n + 1$. The evidence ranking module was found not to be helpful in ranking newly retrieved evidence, often because it strayed too far from the original claim.[1]

The next hop prediction module is implemented by a T5 base sized model (Raffel et al., 2020). T5 consists of a text-to-text encoder-decoder transformer architecture, and its pre-training mixes mul-

[1]We also tried running the evidence ranking model after locating a bridge sentence based on overlap and prepending it to the candidates.

tiple unsupervised objectives on the Colossal Clean Crawled Corpus with supervised NLU tasks including abstractive summarization, question answering, GLUE text classification, and translation, cast into a text to text format. We train the model for three epochs on maximum sequence length 512, using Huggingface default parameters (Wolf et al., 2020). In our task, each input begins with the task identifier "missing: " and a list of the pages retrieved already, followed by the string `[HYP]` and then the claim being classified. Then the elements of the current evidence set (each beginning with a page title in brackets) are concatenated.

Training is based on the gold evidence chains in the training set, and the set of documents retrieved by the baseline model. Every example with evidence from a missing document is used as an example, with the current evidence set being the gold evidence in the retrieved documents and the target evidence being the first piece of evidence from a missing document. For half of the remaining examples (those with no missing documents) including all NEI examples with multiple pieces of evidence, a piece of evidence is randomly left out from the current evidence set, and that evidence is to be predicted as the target. In the other examples, the word "none" is to be predicted, indicating that the evidence chain is complete.

The target output strings are the word "supports" or "refutes," followed by the target evidence in the usual format or "none." For NEI examples, "supports" is to be predicted, indicating a partial evidence chain with no contradictions yet. Thus the log likelihood objective on the target output string amounts to a multi-task objective, combining a prediction of missing evidence with a prediction of the label based on partial information. Because missing evidence should be helpful for label prediction, we hope that co-training on the task of label prediction improves the features used to generate the missing evidence.

The existence of distracting evidence distinguishes the training setting from the testing setting. At test time, the module is always queried with a full set of five sentences and 25 cells, some of which may be irrelevant. For comparison, we trained a model with extracted evidence instead of gold evidence, but the model trained on gold chains achieved more complete chains in fewer hops.

Table 3 describes the performance of the next hop predictor on the development set. "Improved," "Same," and "Worse" count the number of examples where the number of pieces of gold evidence successfully predicted increased, stayed the same, or decreased compared to the previous hop. "Complete" indicates the number of examples for which a complete evidence set is predicted. "FEVEROUS score" is the downstream result of the label classification module (see next section) based on the evidence predicted. Each subsequent hop (up to five) improves the fraction of evidence retrieved, and the FEVEROUS score is monotonically improving up to at least seven hops. This implies that the module knows when to stop and output "none," or else its predictions would eventually overwrite needed evidence from the initial retrieval.

An example of next hop prediction is given in the appendix.

5 Label classification

After the next hop predictor has been run for seven hops, our system uses a label classification module to predict the final label. Another T5 base model is used for this problem, but here we train on the extracted evidence sets (including irrelevant evidence, and missing some gold evidence) that are collected for the training set. Input strings are the same as for the next hop predictor module. The target strings are just "supports," "refutes," or "neutral." As NEI instances only make up 3% of the training set, this label is never learned and the outputs are either "supports" or "refutes."[2]

The label accuracy of this approach on the development set is compared to other approaches that are trained with gold evidence or a RoBERTa model in Table 4. We see that a RoBERTa model has trouble learning in the presence of irrelevant evidence, but is confused by the distractions if only trained on gold evidence chains. In contrast, a T5 model can train and perform successfully on real extracted evidence chains. Consistent with our observations, Jiang et al. (2021) recently established a new state of the art on FEVER using T5 trained on lists of real extracted evidence.

Math hints. As numbers are represented as (possibly several) strings of digits, each with its own pre-trained embedding, it is difficult for the model to answer numerical comparison questions. Also, the model may not precisely know the relationship between a number as a word ("fourteen") and its

[2]In the training set we assign "supports" labels to NEI instances. See below.

Hops	Changes	Improved	Same	Worse	Complete Evidence	FEVEROUS Score
1	—	—	—	—	2661	.271
2	1245	249	7581	60	2722	.276
3	572	77	7768	45	2737	.280
4	391	44	7811	35	2745	.280
5	271	19	7835	36	2748	.281
6	202	13	7846	31	2744	.281
7	166	11	7861	18	2745	.281

Table 3: Performance of the next hop prediction module. FEVEROUS score is based on applying the downstream label classification module after the given hop.

Model	Train/Dev	Label accuracy
RoBERTa	Gold on Gold	.829
RoBERTa	Gold on Extracted	.550
RoBERTa	Extracted on Extracted	.495
T5	Gold on Gold	.848
T5	Gold on Extracted	.572
T5	Extracted on Extracted	.661
T5	Extracted+Math on Extracted+Math	.658

Table 4: Label classification models.

Truth	Supports	NEI	Refutes
Supports	.3403	.5179	.1418
NEI	.0918	.7146	.1936
Refutes	.0822	.4559	.4619
Supports	.6471	.0000	.3529
NEI	.4431	.0000	.5569
Refutes	.2341	.0000	.7659

Table 5: Confusion (development set) when training with (top) and without (bottom) extracted NEI labels.

numerical form ("14").

We attach hints to the beginning of each premise (list of concatenated evidence) as follows. Numbers in the claim or premise appearing in word form (up to twenty, and multiples of ten, one hundred, and one thousand) are converted to their numerical form, and we attach strings such as "four equals 4" for each conversion. Then we collect all numbers (including decimals and integers with commas) with a regular expression, and sort them (along with the number words) from least to greatest, forming a string such as "LEAST 0 less than 1 less than 30 less than 2017 GREATEST". After these prefixes, the original premise begins. It can be clearly recognized because it begins with a title inside brackets.

The NEI class. The NEI class did not have enough examples to be learned reliably in the standard training procedure, but represents 19% of examples in the final test set. To address this, the baseline system upsampled the NEI class by leaving out sentences or entire tables from gold evidence chains to create more NEI examples. For our system, our training data consists of extracted evidence chains rather than gold evidence chains. In addition to the natural NEI examples, we labeled any extracted chain that was still missing information as NEI, gave other extracted chains that were complete their original "supports" or "refutes" label, and trained a T5 base model with the resulting labels. In the resulting training set, 58% of examples were NEI, 20% were refutes, and 23% were supports.

As seen in the confusion matrix of Table 5, the T5 model could not learn the NEI class well and was biased towards NEI even on supporting or refuting examples. Even if 19% of true labels were NEI, as in the test set, the decrease in accuracy on supporting and refuting classes is too great to justify trying to predict this label. Therefore our submitted system is trained to predict only "supports" or "refutes" and never NEI.

An interesting alternative would be to use the ex-

istence of an evidence prediction from the next hop predictor after the final hop to indicate whether an example should be NEI. Following this approach, only 4.4% of NEI examples would be predicted as NEI, compared to 2.8% of supporting and 2.9% or refuting examples, so again including the NEI predictions would yield a net loss.

6 Conclusion

Team Papelo's system for FEVEROUS achieves .281 FEVEROUS score on the development set, with .658 label accuracy and .348 evidence recall. The largest increase in performance over the baseline comes from the label classifier, which uses a different model architecture and is trained on extracted evidence chains including irrelevant evidence. We also achieve better evidence recall through our table cell ranking module, which was trained with a multiple choice cross entropy loss similar to DPR. Additional gains are achieved by our multi-hop evidence retrieval. These modules can only be effective when given good representations of the context of sentences, list items and table cells, which we have carefully constructed.

On the test set we achieve a slightly lower .259 FEVEROUS score. This is largely due to the decrease of label accuracy to .576, reflecting an introduction of an additional 13% of NEI examples compared to the development set (Aly et al., 2021), which our system will always misclassify. The evidence recall of .346 is comparable to the development set.

Already the next hop predictor establishes a beneficial enhancement to the original evidence and can be safely run for many hops. The use of word overlap to match the imagined evidence to actual page elements was a compromise for faster and easier development. We believe the same basic method could be made stronger if a new ranking module, with a similar architecture and training procedure to the preliminary evidence retriever, were trained to match imagined evidence to actually missing evidence. The potential for improvement here is suggested by the number of attempted changes in Table 3, which is always several times the number of evidence sets that were improved.

Additional work is needed to improve performance on particular kinds of examples. Many claims require a system to count certain pieces of retrieved evidence. This skill is taught by datasets such as DROP (Dua et al., 2019) and until recently, neural module networks have needed a stronger form of supervision to learn it (Gupta et al., 2020). A recent alternative (Saha et al., 2021) learns a neural module network with weaker supervision, but instead relies on dependency parsing of the query. To address discrete reasoning examples in FEVEROUS, it may be necessary to integrate models trained on external datasets.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: fact extraction and verification over unstructured and structured information. *CoRR*, abs/2106.05707.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2020. Neural module networks for reasoning over text. In *International Conference on Learning Representations*.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring listwise evidence reasoning with t5 for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 402–410, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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. *CoRR*, abs/1907.11692.

Christopher Malon. 2018. Team papelo: Transformer networks at FEVER. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 109–113, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Amrita Saha, Shafiq R. Joty, and Steven C. H. Hoi. 2021. Weakly supervised neuro-symbolic module networks for numerical reasoning. *CoRR*, abs/2101.11802.

Michael Sejr Schlichtkrull, Vladimir Karpukhin, Barlas Oguz, Mike Lewis, Wen-tau Yih, and Sebastian Riedel. 2020. Joint verification and reranking for open fact checking over tables. *CoRR*, abs/2012.15115.

James Thorne and Andreas Vlachos. 2021. Evidence-based factual error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3298–3309, Online. 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.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

A Example of next hop prediction

Table 6 shows an example where a complete evidence chain is retrieved after 7 hops. The large number of hops is needed because the top-ranked supplementary evidence does not contain the missing information. The imagined needed evidence stays the same until satisfactory evidence is retrieved (after exhausting the higher-ranked evidence) in hop 6. Then the next imagined evidence addresses another part of the reasoning chain. With that, contradictory supplementary evidence is retrieved successfully (northwest versus southwest) and the label for the whole claim is fully supported. Although all five initially retrieved sentences have been replaced before this hop, they are not needed.

Once an example has a complete reasoning chain, its retrieval usually stops long before the seventh hop, by predicting no imagined evidence.

B Example of math hints

Table 7 gives an example of a claim correctly classified with math hints but not without. Although math hints improved some examples, overall label accuracy decreased slightly, perhaps because the length of the hints could push necessary evidence beyond the 512 tokens read by the label classifier.

Claim	Cann River, a river that descends 1,080 metres (3,540 ft) over its 102 kilometres (63 mi) course rises northwest of Granite Mountain and is traversed by the Monaro Highway (which also parallels the former Bombala railway line in several locations) in its upper reaches.
Label	REFUTES
Ground Truth Evidence	[Cann River] The Cann River rises southwest of Granite Mountain in remote country on the eastern boundary of the Errinundra National Park and flows generally east, then south, then east, then south through the western edge of the Coopracambra National Park and through the Croajingolong National Park, joined by seventeen minor tributaries before reaching its mouth with Bass Strait, at the Tamboon Inlet in the Shire of East Gippsland.
Hop 2 Imagined	[Monaro Highway] The Monaro Highway parallels the former Bombala railway line in several locations.
Hop 2 Retrieved	[Monaro Highway] In 1958, it was named the Monaro Highway in both NSW and the ACT, though the same name had been in use by the Snowy Mountains Highway until 1955. *(also two cell retrievals)*
Hop 3 Imagined	[Monaro Highway] The Monaro Highway parallels the former Bombala railway line in several locations.
Hop 4 Imagined	[Monaro Highway] The Monaro Highway parallels the former Bombala railway line in several locations.
Hop 5 Imagined	[Monaro Highway] The Monaro Highway parallels the former Bombala railway line in several locations.
Hop 6 Imagined	[Monaro Highway] The Monaro Highway parallels the former Bombala railway line in several locations.
Hop 6 Retrieved	[Monaro Highway] The road also parallels the former Bombala railway line in several locations. *(also two cell retrievals)*
Hop 7 Imagined	[Cann River] The Cann River rises northwest of Granite Mountain and is traversed by the Monaro Highway in its upper reaches.
Hop 7 Retrieved	[Cann River] The Cann River rises southwest of Granite Mountain in remote country on the eastern boundary of the Errinundra National Park and flows generally east, then south, then east, then south through the western edge of the Coopracambra National Park and through the Croajingolong National Park, joined by seventeen minor tributaries before reaching its mouth with Bass Strait, at the Tamboon Inlet in the Shire of East Gippsland. *(also two cell retrievals)*

Table 6: An example where full evidence is retrieved in seven hops.

Claim	Lamba Kheda recorded a total population of less than 3,000 with 1,100 scheduled castes in the 2011 census.
Label	REFUTES
Premise	LEAST 0.4 less than 2.6 less than 2.7 less than 6 less than 6.25 less than 7.4 less than 8.5 less than 8.7 less than 19.7 less than 28.8 less than 43.1 less than 61 less than 62 less than 82.5 less than 89.5 less than 123 less than 235 less than 289 less than 524 less than 540 less than 560 less than 1100 less than 1850 less than 1977 less than 1981 less than 2011 less than 2058 less than 3000 less than 3166 less than 3908 less than 482365 GREATEST [List of Scheduled Tribes in India] This list has been updated by the Ministry of Tribal Affairs, Government of India, to add the following three. . . . [Lamba Kheda] CAPTION Demographics (2011 Census) KEY VALUE Scheduled caste {{ KEY Total VALUE 1100 }} . . . [Lamba Kheda] VALUE Total {{ KEY Population (2011) VALUE 3,908 }} . . .

Table 7: An example correctly classified using math hints that was misclassified without them.