RIET Lab at CheckThat! 2022: Improving Decoder based Re-ranking for Claim Matching

Michael Shliselberg, Shiri Dori-Hacohen

University of Connecticut

Abstract

The speed and scale at which information can be created and propagated has increased markedly over the last few decades and has far exceeded the scope that human fact-checkers can handle, enabling the rise in harmful and compelling disinformation campaigns. As a result, increasing attention is focusing on the "Claim Matching" problem, where a portion of the fact checking process is automated via AI, by matching content with a human verified claims database. In this report, we discuss a novel neural pipeline for claim matching. Specifically, we demonstrate the efficacy of generative re-rankers to aid the claim matching process, and introduce a new training objective that targets maximizing mutual information. In the CLEF CheckThat! 2022 Competition sub-task 2a, our claim matching approach placed first, beating the second place team by over 3.4 percentage points (evaluated on MAP@5).

Keywords

Re-Ranking, Fact Checking, Generative Language Modeling

1. Introduction

The speed and scale at which information can be created and propagated has increased markedly over the last few decades, due to the rise of internet-based content creation and social media. Along with the many benefits of this increase, it has also enabled malicious actors to initiate disinformation campaigns [1] with real world implications, shaping public opinion in forms of politics, pandemics, and more. Human fact checking efforts, while usually high-quality (i.e. high precision) and impactful, fail to match false information's reach, speed and scale [2]. To remedy this, researchers around the world have begun to utilize Machine Learning in order to automate portions of the fact checking process, with the goal of scaling and speeding up these efforts to match the strong need. However, many challenges exist when fact checking itself becomes automated or determination of veracity is left in the hands of opaque machines. Therefore, one of the leading approaches relies on a specific subproblem, commonly referred to as "Claim Matching", whereby machine learning is used to map novel content, such as social media posts, with a database of fact-checked claims, i.e., claims that have already been verified to be true or false [3, 4]. Claim matching approaches can work well either as standalone systems, or else integrated into human in-the-loop systems, where the automated portion can be used to triage and prioritize incoming posts/claims, and/or to point users to existing fact checks, e.g. via fact checking tiplines [5].

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

michael.shliselberg@uconn.edu (M. Shliselberg); shiridh@uconn.edu (S. Dori-Hacohen)

Current literature on claim matching focuses on encoding-based methods [6, 7]. Most of these approaches are built with deep pre-trained language models. These models have become ubiquitous in natural language processing since the emergence of BERT [7] in 2018. In addition to encoding-based approaches, generative models have recently received renewed interest due to their ability to provide further insight and understanding of the model's inferences [8]. To the best of our knowledge, we have not seen deep generative models used specifically within the subdomain of claim matching.

2. Data

For evaluation we use CLEF's CheckThat! task 2a-English claim-matching competition data from 2021 and 2022 [9, 10]. The data-sets comprise of tweets, a verified natural-language claims database, and their corresponding connections, with each tweet mapped to precisely one claim. The claims database has approximately 14000 claims. Each year, CLEF's CheckThat! team provides 1000 training points, 200 development points, and around 200 for test. Note that at the time of writing, we have access only to the test labels from 2021's competitions, but not for 2022; our results are quoted from the official, finalized leaderboard.

3. Candidate Selection

Our claim matching pipeline is broken down into two stages following the traditional neural IR methodology of first using an efficient model for candidate selection, and then following up with a more expensive model to reevaluate those candidates. More specifically, we will consider methods that run in $\mathcal{O}(C)$ to be efficient, where C is the size of the claim database. For candidate selection, we test 2 approaches. The first is a lightweight bag-of-words approach which only utilizes unigram distributions, and the second uses a sentence transformer, which is still linear, but models context via deep neural networks. We will discuss each in turn, before introducing the re-ranking approach in the following section.

3.1. Bag of Words

We use BM25 for the bag of words baseline, implemented by the rank-bm25 python package. We preprocess the text in 3 steps: (1) we concatenate the claim's title, header, and body (see, e.g., [11]); (2) we convert all text to lowercase and remove stop words, using NLTK's English list [12]; and finally (3) and we apply Porter stemming.

3.2. Sentence Transformer

Sentence transformers [13] utilize the widely successful self-attention based architectures [14], but are trained to produce embeddings projected to a unit sphere in Euclidean space. This means that computing the angle (or Euclidean distance) between two inputs can have contextual meaning, enabling search with $\mathcal{O}(T + C)$ inferences; the alternative of computing each tweet-claim pair directly requires $\mathcal{O}(T * C)$ runs, where T is the number of tweets. Specifically, we

use Sentence-T5 [15], which to the best of our knowledge is the strongest sentence transformer when normalized by model size.

To best incorporate batched training, we use a Multiple Negative Rankings (MNR) loss function. MNR's batched loss is described by

$$\mathcal{L}(\mathcal{B}) = \sum_{(x,y)\in\mathcal{B}} -\log \frac{S(x,y)}{\sum_{(\_,y^*)\in\mathcal{B}} S(x,y^*)} \tag{1}$$

Where, given a network f, $S(x,y)$ is the function $exp(f(x)^T f(y))$. This loss function is appealing for efficiency reasons: it can be computed with a single training step using inter-batch normalization. In our case, we also wanted to add "harder" negatives because otherwise the task becomes too easy for the model. In other words, we have $\mathcal{B} = \mathcal{B}^+ \cup \mathcal{B}^-$, and the loss becomes

$$\mathcal{L}(\mathcal{B}) = \sum_{(x,y)\in\mathcal{B}^+} -\log \frac{S(x,y)}{\sum_{(\_,y^*)\in\mathcal{B}} S(x,y^*)} \tag{2}$$

To determine the hard negatives, we use our BM25 Model to create ranked lists, and take the top ranked negative tweet.

4. Generative Re-ranking

The re-ranking task is posed as: given a tweet t and a set of claim candidates C_{cs}, return a ranked list of the claims. As discussed above, the generative approach models $p(t|c)$ and uses that probability to generate the ranked list.

We decided to use a full-decoder setup, given the recent success of models like GPT-3 and Jurassic [16, 17], who have shown numerous state-of-the-art results in low-data settings. For the backbone architecture, we use GPT-Neo's 1.3 Billion parameter model [18], which is a large auto-regressive transformer trained on the Pile [19]. We leave most of the architecture as-is, only modifying the model's usage of positional embeddings. In our case, we reset the ids for the claim and tweet separately.

To use the full decoder setting, we need to convert our inputs into a prompt. We use a straightforward setup which takes as input a claim c and a tweet t, and turn it into the prompt $< bos > Claim : \{c\} < eos >< bos > Tweet : \{t\} < eos >$, where $< bos >$ and $< eos >$ represent the model tokenizer's native beginning-of-sentence and end-of-sentence tokens respectively.

4.1. Training Objective

While most approaches that leverage generative models for re-ranking rely on negative examples [20], we introduce a new method that does not. We do so by taking advantage of a full decoder's setup to also model the priors of the tweet, $p(t)$. This allows us to create a retriever-agnostic model, which requires no assumption of negatives. While this capability may be useful in contexts where retrieving negatives are difficult, we show that a fused objective when negatives are available performs the strongest.

Mutual Information is defined as

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \tag{3}$$

$$= \mathbb{E}_{X,Y} \log \frac{P(x|y)}{P(x)} \tag{4}$$

The expectation in the last line can be approximated stochastically in the optimization scheme but the fractional piece would require each batch to be made up of an additional element (prompt(t), prompt(t,c)) $\in \mathcal{B}$. Given network q_θ, the optimization problem becomes

$$\theta^* = arg \min_\theta \mathbb{E}_{C,T}[-\log q_\theta(t|c)) + \log q_\theta(t)] \tag{5}$$

To handle the objective not being taken over by the model weakening certain priors, common practice is to develop a hinged form of the objective. So we create the Hinged Mutual Information (HMI) loss as

$$\mathcal{L}_{HMI} = \mathbb{E}_{C,T}[max(0, \lambda - \log q_\theta(t|c)) + \log q_\theta(t))] \tag{6}$$

This will remove elements with values above threshold λ, which in training regimes that only pass the data at most once, is equivalent to removing them from the training set. To take advantage of the data, we decide to only mask the prior in the setting of surpassing the threshold and revert the loss back to Maximum Likelihood Encoding (MLE). We will denote this new loss as Hinged Prior Mutual Information Loss (HPMI)

$$\mathcal{L}_{HPMI} = \mathbb{E}_{C,T} \begin{cases} -\log q_\theta(t|c) + \log q_\theta(t), & \text{if } -\log \frac{q_\theta(t|c)}{q_\theta(t)} < \lambda \\ -\log q_\theta(t|c), & \text{else} \end{cases} \tag{7}$$

While there is the memory-based drawback that each training element requires 2 runs of the model, one for computing $q_\theta(t|c)$ and one for $q_\theta(t)$, we can take advantage of it by extracting additional regularization using the models Posterior. The only adjustment is that we input $q_\theta(c|t)$ instead of $q_\theta(t)$, requiring a flipped ordered prompt: tweet $\rightarrow$ claim. Given our first run we also inherently get a claim prior $q_\theta(c)$ thanks to the decoder setup, which in tandem with the posterior lets us model mutual information another way by trying to model $\log \frac{p(c|t)}{p(c)}$. Since we care less about claim priors, we regularize our loss by just using its hinged form, which we will denote Posterior based Hinged Mutual Information Loss $\mathcal{L}_{PoHMI}$ which is identical to equation 6 just by replacing likelihood and tweet prior with the posterior and claim prior. Note that we do not care about modeling the claim via MLE in the hinged condition as we do in equation 7, so we just zero it out in those cases. Additionally, this doesn't model the exact same Mutual Information metric as the other losses as it adjusts the random variables positioning, which was another reason we reset the position-ids to help mitigate that discrepancy.

Now that we have explained our retriever free training objective to best utilize vanilla data, since we do have a retriever we also incorporate negatives via negative log likelihood loss (nl3u) [20] getting a final training objective of

$$\mathcal{L}_{mix} = \mathcal{L}_{HPMI} + \mathcal{L}_{PoHMI} + \mathcal{L}_{NL3U} \tag{8}$$

5. Results

Model	MAP@1	MAP@5
aschern	.861	.883
bm25-basic	.634	.688
bm25-prepr	.801	.855
sent-T5	.812	.862
neo-reranked	**.939**	**.96**

Table 1
Results on Test set of CLEF CheckThat! 2021 task 2a Claim Matching dataset. sent-T5 refers to sentence-T5 large model, neo-reranked is GPT-Neo re-ranker, and aschern refers to the 2021 winning submission [11].

Team/User	MAP	MAP@1	MAP@5
Motlogelwan	.8775	.8325	.8731
Simba [21]	.9075	.8756	.9075
Viktor [22]	.9223	.9043	.9222
BigIR [23]	.9225	.8995	.9211
RIET Lab (ours)	**.957**	**.9426**	**.9555**

Table 2
Officially released final Leaderboard for CLEF CheckThat! 2022 task 2a competition.

Model	MAP@1	MAP@5
NL3U [20]	.931	.952
HPMI	.911	.94
HPMI + PoHMI	.92	.942
Mix	**.939**	**.96**

Table 3
Results of different training rewards mentioned in section 4 on the Test set of CLEF CheckThat! 2021 task 2a Claim Matching dataset.

For evaluation, we follow the CLEF competitions leader-board evaluation and use Mean Average Precision (MAP). While they were finally scored on MAP@5, we show results for both MAP@1 and MAP@5. In this setting we evaluate with only 5 candidates.

To compute Average Precision @k (AP@k) for some ranked list R with singular gold label g, it follows this formula

$$AP@k\{R, g\} = \sum_{i=1}^{k} \mathbb{1}[R[i-1] == g] \frac{1}{i} \tag{9}$$

where $\mathbb{1}[\cdot]$ is just an indicator function. MAP@k is just the mean of equation 9 over all ranked lists. Which with set of ranked list and corresponding label pairs denoted as Ω, becomes

$$MAP@k\{\Omega\} = \frac{1}{|\Omega|} \sum_{R,g \in \Omega} AP@k\{R, g\} \tag{10}$$

The results of baselines, encoding methods, and re-ranked versions are in Table 1 where we evaluated on 2021's competition data where the test set is accessible. As seen, re-ranking heavily improved upon it's candidate selection counterpart. Compared to the previous years competition winner that utilized sentence models and probabilistic re-ranking [11], the best model is 6 percentage points stronger. In Table 2 you see the results of 2022's competition's top 5 submissions where we submitted our generative re-ranking pipeline utilizing 25 candidates per tweet. We placed first beating out second place by around 3.4 percentage points.

We also completed a small ablation study on the rewards discussed in section 4, where the results are shown in Table 3. While a negative required loss still beat the ones we proposed where it isn't, it is only by a slight margin. Combining the approaches yielded the best results, and is what we used for our submission within the competition.

6. Conclusion and Future Directions

Claim matching holds significant promise as a component in mitigating mis- and disinformation. In this report, we have introduced a new objective for full-decoder generative re-ranker architectures that do not rely on negative samples as well as a mixed objective. Our approach is competitive; it improves upon the state-of-the-art results in the CLEF-2021 check-that task 2a, and was utilized to place first within the 2022 competition.

In this report, we focus only on using this pipeline for claim matching. Our work points to several directions for future work, such as applying the pipeline to other challenges; and comparing different objective functions, as well as their corresponding data assumptions. Though beyond the scope of this report, we further hypothesize that utilizing this pipeline in a human-in-the-loop system would be valuable, given the access to confidence statistics from the generative approach, which could not be achieved in encoder systems; we leave this study for future work.

6.1. Instructions to Reproduce

For finetuning the Sentence-T5 model we use the AdamW optimizer with a constant learning rate of 5e-6 with batch-size 6 for a single epoch (1 negative per positive in each batch). Training was performed on a single A6000 GPU. For finetuning the re-rankers we also use the AdamW optimizer with a constant learning rate of 2e-5 with batch-size 4 and reward hyperparameter λ of 2 for a single epoch. We performed this portion of training across 4 A6000 GPUs.

Our code can be found at https://github.com/RIET-lab/GenerativeClaimMatchingPipeline.

Acknowledgments

This material is based upon work supported by the National Science Foundation Convergence Accelerator (contract 49100421C0035). Shiri Dori-Hacohen holds a significant financial interest in AuCoDe. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Also a special thanks to Scott Hale from Meedan and Zeeshan Lodhia from the RIET lab for comments, ideas, and support!

References

[1] X. Zhang, A. A. Ghorbani, An overview of online fake news: Characterization, detection, and discussion, Inf. Process. Manag. 57 (2020) 102025.

[2] K. H. Jamieson, D. Romer, P. E. Jamieson, K. M. Winneg, J. Pasek, The role of non–covid-specific and covid-specific factors in predicting a shift in willingness to vaccinate: A panel study, Proceedings of the National Academy of Sciences 118 (2021) e2112266118. URL: https://www.pnas.org/doi/abs/10.1073/pnas.2112266118. doi:10.1073/pnas.2112266118. arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.2112266118.

[3] S. Nieminen, V. Sankari, Checking politifact's fact-checks, Journalism Studies 22 (2021) 358–378. URL: https://doi.org/10.1080/1461670X.2021.1873818. doi:10.1080/1461670X.2021.1873818. arXiv:https://doi.org/10.1080/1461670X.2021.1873818.

[4] Y. Nie, H. Chen, M. Bansal, Combining fact extraction and verification with neural semantic matching networks, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 6859–6866. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4662. doi:10.1609/aaai.v33i01.33016859.

[5] A. Kazemi, K. Garimella, G. K. Shahi, D. Gaffney, S. A. Hale, Tiplines to combat misinformation on encrypted platforms: A case study of the 2019 indian election on whatsapp, CoRR abs/2106.04726 (2021). URL: https://arxiv.org/abs/2106.04726. arXiv:2106.04726.

[6] L. Konstantinovskiy, O. Price, M. Babakar, A. Zubiaga, Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection, CoRR abs/1809.08193 (2018). URL: http://arxiv.org/abs/1809.08193. arXiv:1809.08193.

[7] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[8] O. Lesota, N. Rekabsaz, D. Cohen, K. A. Grasserbauer, C. Eickhoff, M. Schedl, A modern perspective on query likelihood with deep generative retrieval models, CoRR abs/2106.13618 (2021). URL: https://arxiv.org/abs/2106.13618. arXiv:2106.13618.

[9] S. Shaar, F. Haouari, W. Mansour, M. Hasanain, N. Babulkov, F. Alam, G. Da San Martino, T. Elsayed, P. Nakov, Overview of the CLEF-2021 CheckThat! lab task 2 on detecting previously fact-checked claims in tweets and political debates, in: Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum, CLEF '2021, Bucharest, Romania (online), 2021. URL: http://ceur-ws.org/Vol-2936/paper-29.pdf.

[10] P. Nakov, G. Da San Martino, F. Alam, S. Shaar, H. Mubarak, N. Babulkov, Overview of the CLEF-2022 CheckThat! lab task 2 on detecting previously fact-checked claims, in: Working Notes of CLEF 2022—Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[11] A. Chernyavskiy, D. Ilvovsky, P. Nakov, Aschern at checkthat! 2021: normalcr lambda-calculus of fact-checked claims, 2021.

[12] E. Loper, S. Bird, Nltk: The natural language toolkit, CoRR cs.CL/0205028 (2002). URL: http://dblp.uni-trier.de/db/journals/corr/corr0205.html#cs-CL-0205028.
[13] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL: https://arxiv.org/abs/1908.10084. doi:10.48550/ARXIV.1908.10084.
[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. URL: https://arxiv.org/abs/1706.03762. doi:10.48550/ARXIV.1706.03762.
[15] J. Ni, G. H. Ábrego, N. Constant, J. Ma, K. B. Hall, D. Cer, Y. Yang, Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models, 2021. URL: https://arxiv.org/abs/2108.08877. doi:10.48550/ARXIV.2108.08877.
[16] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. URL: https://arxiv.org/abs/2005.14165. doi:10.48550/ARXIV.2005.14165.
[17] O. Lieber, O. Sharir, B. Lenz, Y. Shoham, Jurassic-1: Technical Details And Evaluation, Technical Report, AI21 Labs, 2021.
[18] S. Black, G. Leo, P. Wang, C. Leahy, S. Biderman, GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, 2021. URL: https://doi.org/10.5281/zenodo.5297715. doi:10.5281/zenodo.5297715, If you use this software, please cite it using these metadata.
[19] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al., The pile: An 800gb dataset of diverse text for language modeling, arXiv preprint arXiv:2101.00027 (2020).
[20] C. N. dos Santos, X. Ma, R. Nallapati, Z. Huang, B. Xiang, Beyond [cls] through ranking by generation, in: EMNLP, 2020.
[21] A. Hövelmeyer, K. Boland, S. Dietze, SimBa at CheckThat! 2022: lexical and semantic similarity based detection of verified claims in an unsupervised and supervised way, in: N. Faggioli, Guglielmo andd Ferro, A. Hanbury, M. Potthast (Eds.), Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.
[22] V. Kostov, AI Rational at CheckThat! 2022: reranking previously fact-checked claims on semantic and lexical similarity, in: N. Faggioli, Guglielmo andd Ferro, A. Hanbury, M. Potthast (Eds.), Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.
[23] W. Mansour, T. Elsayed, A. Al-Ali, Did i see it before? detecting previously-checked claims over twitter, in: European Conference on Information Retrieval, Springer, 2022, pp. 367–381.