Rethinking Loss Functions for Fact Verification

Yuta Mukobara[a,†] **Yutaro Shigeto**[b,c,‡] **Masashi Shimbo**[b,c]
[a] Tokyo Institute of Technology [b] STAIR Lab, Chiba Institute of Technology [c] RIKEN AIP
mukobara.y.aa@m.titech.ac.jp {shigeto,shimbo}@stair.center

Abstract

We explore loss functions for fact verification in the FEVER shared task. While the cross-entropy loss is a standard objective for training verdict predictors, it fails to capture the heterogeneity among the FEVER verdict classes. In this paper, we develop two task-specific objectives tailored to FEVER. Experimental results confirm that the proposed objective functions outperform the standard cross-entropy. Performance is further improved when these objectives are combined with simple class weighting, which effectively overcomes the imbalance in the training data. The source code is available.[1]

1 Introduction

The Fact Extraction and VERification (FEVER) shared task (Thorne et al., 2018) challenges systems to verify a given claim by referencing Wikipedia articles. A system for FEVER typically begins by extracting sentences from Wikipedia that potentially support or refute the claim. Subsequently, the verdict predictor in the system classifies the claim, in conjunction with the retrieved sentences, into one of three verdict classes:

- Supported (SUP): The retrieved sentences contain evidence supporting the given claim.
- Refuted (REF): The retrieved sentences contain evidence that refutes the claim.
- Not Enough Information (NEI): The retrieved sentences do not contain sufficient evidence to support or refute the claim.

As this verification step is a multiclass classification task, verdict predictors are usually trained using the cross-entropy loss function. However, cross-entropy treats all misclassification types uniformly, which is problematic given the heterogeneity among the verdict classes in FEVER; labels SUP and REF both assume evidence is present in the retrieved sentences, whereas a claim is deemed NEI only when such evidence is missing. Consequently, it is debatable, for example, whether misclassifying a SUP claim as REF or as NEI should be considered equally severe errors, especially when the retrieved sentences indeed contain supporting evidence, such as when a verdict predictor is trained with oracle sentences.

In this paper, we explore objective functions designed to capture the heterogeneity among verdict classes.

Notation For a K-class classification problem, let $\mathbf{y} = (y_1, \ldots, y_K) \in \{0,1\}^K$ denote a one-hot class representation vector where each index represents a class. Depending on the context, we also use $\mathbf{y}$ to denote the corresponding class itself. Let $\mathbf{p} = (p_1, \ldots, p_K) \in [0,1]^K$ denote a predicted class distribution (i.e., $\sum_{i=1}^K p_i = 1$). For FEVER verdict prediction, $K = 3$, and let the indexes $1, 2, 3$ correspond to SUP, REF, NEI, respectively.

2 Proposed Method

2.1 Cross-entropy Loss Function

We first review the (categorical) cross-entropy loss, which is a common objective function for multiclass classification, including FEVER verdict prediction (Liu et al., 2020; Tymoshenko and Moschitti, 2021).

In a K-class classification task, the cross-entropy loss for a sample with its one-hot class vector $\mathbf{y} = (y_1, \ldots, y_K)$ is defined as:

$$L_{\mathrm{CE}}(\mathbf{y}, \mathbf{p}) = -\sum_{i=1}^{K} y_i \log p_i, \tag{1}$$

where $\mathbf{p} = (p_1, \ldots, p_K)$ is the class probability

[†] Work conducted during an internship at STAIR Lab.
[‡] Corresponding author.
[1] https://github.com/yuta-mukobara/RLF-KGAT

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 2: Short Papers, pages 432–442

distribution derived from the output of a classifier through a softmax function.

2.2 Loss Functions for Verdict Prediction

To address the heterogeneity of verdict classes outlined in Section 1, we implement penalties of varying magnitudes contingent on the type of prediction errors. To be precise, our objectives impose more severe penalties for incorrectly classifying SUP claims as REF, or REF claims as SUP, considering that classes SUP and REF are contradictory when the retrieved sentences contain correct evidence. Note that this last condition is constantly met during training with oracle sentences in the FEVER dataset.

2.2.1 Multi-label logistic loss

Before presenting our loss functions for FEVER, we introduce the multi-label logistic (MLL) loss (Baum and Wilczek, 1988). Although this loss is not suited for FEVER verdict prediction, its inclusion of loss terms for complementary classes helps illustrate our approach.

The MLL loss is defined as the sum of logistic losses (binary cross-entropy) over K components of the predictor's output $\mathbf{p}$:

$$\begin{aligned} L_{\mathrm{MLL}}(\mathbf{y}, \mathbf{p}) &= -\sum_{i=1}^{K}[y_i \log p_i \quad \lambda \bar{y}_i \log(1 - p_i)], \\ &= L_{\mathrm{CE}}(\mathbf{y}, \mathbf{p}) \quad \lambda R_{\mathrm{MLL}}(\mathbf{y}, \mathbf{p}) \end{aligned} \tag{2}$$

where:

$$R_{\mathrm{MLL}}(\mathbf{y}, \mathbf{p}) = -\sum_{i=1}^{K} \bar{y}_i \log(1 - p_i), \tag{3}$$

with $\bar{y}_i = 1 - y_i$. As Eq. (2) shows, the MLL loss consists of the primary cross-entropy term and an auxiliary term R_{MLL} for complementary classes. Also note that, in the original MLL loss, $\lambda = 1$, but we treat $\lambda \geq 0$ as a hyperparameter that can also take a different value to control the balance between two terms.

Originally, since the MLL loss was designed for multi-label classification, the K outputs of a predictor are treated as independent variables. Therefore, each component of the prediction vector $\mathbf{p}$ is independently normalized using the sigmoid function. In contrast, within the scope of this paper, $\mathbf{p}$ forms a probability distribution via the softmax function, suitable for a multi-class setting of FEVER.

One interpretation of this loss is that the predicted class distribution $\mathbf{p} = (p_1, \ldots, p_K)$ is viewed not as the outcome of a single K-class classification task, but as the outcomes of K "one-versus-rest" binary classification tasks; in each of these tasks, one of the K classes is treated as the positive class, while the remaining $K - 1$ classes are treated collectively as the negative class, and then individual tasks evaluated by the logistic loss.

Application to verdict prediction In Eqs. (2) and (3), $\bar{y}_i = 1 - p_i$ indicates the membership of the ith class in the complement of class $\mathbf{y}$, i.e., in the set $Y \setminus \{\mathbf{y}\}$. In the context of FEVER, the complement sets for individual verdict classes are $\overline{\text{SUP}} = \{\text{REF}, \text{NEI}\}$, $\overline{\text{REF}} = \{\text{SUP}, \text{NEI}\}$, and $\overline{\text{NEI}} = \{\text{SUP}, \text{REF}\}$. Now, setting $K = 3$ and recalling that class indexes $1, 2, 3$ represent SUP, REF, NEI, respectively, we have:

$$\begin{aligned} &R_{\mathrm{MLL}}(\mathbf{y}, \mathbf{p}) \\ &\quad = \begin{cases} -\log(1 - p_2) - \log(1 - p_3), & \text{if } y_1 = 1, \\ -\log(1 - p_3) - \log(1 - p_1), & \text{if } y_2 = 1, \\ -\log(1 - p_1) - \log(1 - p_2), & \text{if } y_3 = 1. \end{cases} \end{aligned} \tag{4}$$

Eq. (4) is symmetric over classes, which shows that the MLL loss does not account for the heterogeneity among verdict classes, much like the cross-entropy loss. Later experiments in Section 3 indeed demonstrate that the MLL loss does not improve over the standard cross-entropy in terms of prediction accuracy.

2.2.2 Reducing penalties for false NEI

We address the issue of heterogeneous verdict classes by modifying the composition of complement sets in the MLL loss.

Specifically, in our first FEVER-specific loss function, we treat classes SUP and REF as their sole complementary class, excluding NEI. To be precise, we let $\overline{\text{SUP}} = \{\text{REF}\}$, $\overline{\text{REF}} = \{\text{SUP}\}$, whereas $\overline{\text{NEI}} = \{\text{SUP}, \text{REF}\}$ is unchanged. Accordingly, the membership indicator $\bar{y}_i$ is changed to:

$$\bar{y}_i^{\mathrm{SRN}} = \begin{cases} 1 - y_i, & \text{if } i = 1, 2, \\ 0, & \text{if } i = 3, \end{cases} \tag{5}$$

which results in:

$$\begin{aligned} R_{\mathrm{SRN}}(\mathbf{y}, \mathbf{p}) &= -\sum_{i=1}^{3} \bar{y}_i^{\mathrm{SRN}} \log(1 - p_i) \\ &= -\sum_{i=1}^{2}(1 - y_i) \log(1 - p_i) \end{aligned}$$

$$
= \begin{cases} -\log(1-p_2), & \text{if } y_1 = 1, \\ -\log(1-p_1), & \text{if } y_2 = 1, \\ -\log(1-p_1) - \log(1-p_2), & \text{if } y_3 = 1. \end{cases} \tag{6}
$$

Comparing the last formula with Eq. (4), we see that R_{SRN} effectively reduces penalties for misclassifying SUP or REF claims (i.e., $y_1 = 1$ or $y_2 = 1$) as NEI. Combining the auxiliary loss with the cross entropy loss, we obtain the overall objective:

$$
\begin{aligned}
L_{\text{SRN}}(\mathbf{y}, \mathbf{p}) &= L_{\text{CE}}(\mathbf{y}, \mathbf{p}) \quad \lambda R_{\text{SRN}}(\mathbf{y}, \mathbf{p}) \\
&= -\sum_{i=1}^{3} y_i \log p_i - \lambda \sum_{i=1}^{2} (1-y_i) \log(1-p_i).
\end{aligned} \tag{7}
$$

2.2.3 Exclusive penalties for SUP/REF confusion

Alternatively, we can define an auxiliary loss focusing only on the contradictory nature of SUP and REF and disregarding NEI entirely. To this end, we define $\overline{\text{NEI}} = \emptyset$. For SUP and REF, their complementary sets are defined in the same way as the SRN loss term, namely, $\overline{\text{SUP}} = \{\text{REF}\}$ and $\overline{\text{REF}} = \{\text{SUP}\}$. The corresponding membership indicator is given by:

$$
\bar{y}_i^{\text{SR}} = \begin{cases} (1-y_i)(1-y_3), & \text{if } i = 1, 2, \\ 0, & \text{if } i = 3. \end{cases}
$$

The newly introduced factor $(1 - y_3)$ ensures $\bar{y}_i^{\text{SR}}$ remains 0 when the gold label is NEI (and thus $y_3 = 1$). This produces our second auxiliary loss function for FEVER:

$$
\begin{aligned}
R_{\text{SR}}(\mathbf{y}, \mathbf{p}) &= -\sum_{i=1}^{3} \bar{y}_i^{\text{SR}} \log(1-p_i) \\
&= -(1-y_3) \sum_{i=1}^{2} (1-y_i) \log(1-p_i) \\
&= \begin{cases} -\log(1-p_2), & \text{if } y_1 = 1, \\ -\log(1-p_1), & \text{if } y_2 = 1, \\ 0, & \text{if } y_3 = 1. \end{cases}
\end{aligned} \tag{8}
$$

In this loss term, any misclassification involving label NEI is disregarded; R_{SR} imposes no penalty for prediction errors on NEI claims, nor for misclassifying SUP and REF claims as NEI.

The overall objective function, combining R_{SR} with L_{CE}, is given as follows:

$$
\begin{aligned}
L_{\text{SR}}(\mathbf{y}, \mathbf{p}) &= L_{\text{CE}}(\mathbf{y}, \mathbf{p}) \quad \lambda R_{\text{SR}}(\mathbf{y}, \mathbf{p}) \\
&= -\sum_{i=1}^{3} y_i \log p_i \\
&\quad - \lambda (1-y_3) \sum_{i=1}^{2} (1-y_i) \log(1-p_i).
\end{aligned} \tag{9}
$$

2.3 Class Imbalanced Learning

Another non-negligible issue in verdict prediction is the imbalanced training data in the FEVER dataset, whose class frequency is shown in Table 1.

A popular approach to class imbalance problems (Zhang et al., 2023; Chawla et al., 2002) is class weighting (Ren et al., 2018; Cui et al., 2019), where each term in the objective function is assigned a different weight depending on the class it is associated with.

For example, after weighting applied, the SRN objective in Eq. (7) becomes:

$$
\begin{aligned}
&L_{\text{SRN\ weighting}}(\mathbf{y}, \mathbf{p}) \\
&= -\sum_{i=1}^{3} w_i \left[y_i \log p_i \quad \bar{y}_i^{\text{SRN}} \log(1-p_i) \right],
\end{aligned} \tag{10}
$$

where w_1, w_2, and w_3 are the fixed class weights. The same weighting scheme can be applied to SR and MLL objective functions; see Appendix A.

In our experiments in Section 3, we use the class-balanced weights of Cui et al. (2019). They define the weight for the ith class as:

$$
w_i = \frac{1-\beta}{1-\beta^{n_i}}, \tag{11}
$$

where n_i is the number of training samples in the ith class and β is a hyperparameter. Setting $\beta = 0$ results in uniform weights $w_1 = w_2 = w_3 = 1$, which reduces Eq. (10) to the unweighted one in Eq. (7). As $\beta \to 1$, the weights approach the inverse class frequency $1/n_i$.

3 Experiments

Due to limited space, only the main experimental results are presented below. Additional results and analysis can be found in Appendix B.

3.1 Setups

Dataset and evaluation criteria The FEVER 2018 dataset (Thorne et al., 2018) consists of 185,445 claims (Table 1). Each claim is assigned a gold class labels, SUP, REF, or NEI. The gold labels for the test set are not disclosed.

Models are evaluated by prediction label accuracy (LA) and FEVER score (FS). LA is a standard

Split	#SUP	#REF	#NEI
Train	80,035	29,775	35,639
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 1: Number of samples (claim-evidence pairs) in the FEVER 2018 dataset.

evaluation criterion for multiclass classification where classification accuracy is computed without considering the correctness of the retrieved evidence. In FS, a prediction is deemed correct only if the predicted label is correct and the correct evidence is retrieved (in the case of SUP and REF claims). The scores for the test set, for which the gold labels are not disclosed, are computed on the official FEVER scoring site.

Compared models and hyperparameters We use KGAT[2] (Liu et al., 2020) for both evidence retrieval and verdict prediction. Multiple prediction models are trained, each with a different objective function. The objectives employed are:

- CE: The cross-entropy loss of Eq. (1). This is the standard objective function for FEVER. It is used by the original KGAT, and is the baseline in our experiments.
- MLL: The multi-label logistic loss of Eq. (2). As our proposed objectives can be considered its modifications, it is included as another baseline in this comparative study.
- SRN: Our first proposed objective (Eq. (7)), which combines the cross-entropy loss with the R_{SRN} auxiliary loss.
- SR: Our second proposed objective (Eq. (9)), which augments the cross-entropy loss with the R_{SR} auxiliary loss.

Each objective is assessed with and without the class weighting scheme of Eq. (11). A summary of all objective functions evaluated can be found in Appendix A. Additionally, all objectives are evaluated with three different backbone networks: BERT Base, BERT Large (Devlin et al., 2019), and RoBERTa Large (Liu et al., 2019).

Hyperparameters λ in Eqs. (2), (7), and (9), and β in Eq. (11) are tuned on the development set. For other hyperparameters (e.g., learning rate and batch size), the default values set in the KGAT implementation are used. Each model is trained three times and the one achieving the highest LA on the development set is selected for evaluation.

[2] https://github.com/thunlp/KernelGAT

Objective function	Weighting	LA	FS
Backbone: BERT Base			
CE (baseline)	–	77.81	75.75
CE	yes	78.08 (+0.27)	76.02 (+0.27)
MLL (λ=0.0625)	–	77.84 (+0.03)	75.65 (-0.10)
MLL (λ=0.125)	yes	78.13 (+0.32)	**76.06 (+0.31)**
SRN (λ=0.0625)	–	77.84 (+0.03)	75.70 (-0.05)
SRN (λ=0.0625)	yes	77.83 (+0.02)	75.79 (+0.04)
SR (λ=0.0625)	–	78.16 (+0.35)	75.87 (+0.12)
SR (λ=0.25)	yes	**78.29 (+0.48)***	**76.06 (+0.31)**
Backbone: BERT Large			
CE (baseline)	–	78.20	75.98
CE	yes	78.85 (+0.65)*	76.74 (+0.76)
MLL (λ=0.25)	–	78.94 (+0.74)*	76.78 (+0.80)
MLL (λ=0.03125)	yes	78.85 (+0.65)*	76.74 (+0.76)
SRN (λ=0.125)	–	78.68 (+0.48)*	76.57 (+0.59)
SRN (λ=0.25)	yes	78.83 (+0.63)*	76.71 (+0.73)
SR (λ=0.25)	–	79.02 (+0.82)*	76.86 (+0.88)
SR (λ=0.125)	yes	**79.19 (+0.99)***	**77.01 (+1.03)**
Backbone: RoBERTa Large			
CE (baseline)	–	80.19	78.03
CE	yes	80.55 (+0.36)	78.54 (+0.51)
MLL (λ=0.0625)	–	80.00 (-0.19)	77.88 (-0.15)
MLL (λ=0.0625)	yes	80.62 (+0.43)*	78.55 (+0.52)
SRN (λ=0.03125)	–	80.24 (+0.05)	78.18 (+0.15)
SRN (λ=0.03125)	yes	**80.73 (+0.54)***	78.56 (+0.53)
SR (λ=0.0625)	–	80.41 (+0.22)	78.19 (+0.16)
SR (λ=0.03125)	yes	80.70 (+0.51)*	**78.63 (+0.60)**

Table 2: Label accuracy (LA) and FEVER score (FS) of KGAT models on the development set, using different loss functions and backbones. For class-balanced weighting, β is set to 0.999999 in all cases. The parenthesized figures after LA indicate differences from the baseline cross-entropy loss (CE) without class-balanced weighting. Asterisks (*) denote the change in prediction from CE (baseline) is statistically significant ($p < 0.05$), as determined by the McNemar test (McNemar, 1947).

3.2 Results

Effectiveness of the proposed objective functions Table 2 shows the results. Trends observed are: (i) The imbalance weighting consistently improves both LA and FS. (ii) The proposed SRN and SR losses enhance LA in all cases and FS in most cases. (iii) The simultaneous use of the class-balance weighting and the proposed losses further improves the performance.

Of the two proposed loss types, SR achieves higher scores across all backbone architectures, with the exception of the LA score with RoBERTa Large. Even in the latter case, the difference is marginal (0.03). For SR with weighting, the change in predictions from CE (baseline) is statistically significant irrespective of the backbones. The same is true for SRN with weighting, except when it is

Method	Dev LA	Dev FS	Test LA	Test FS
Backbone: BERT Base				
KGAT (Liu et al., 2020)	78.02	75.88	72.81	69.40
KGAT (reproduced)	77.81	75.75	73.01	69.29
KGAT + SR + weighting	**78.29**	**76.06**	**73.44**	**69.88**
Backbone: BERT Large				
KGAT (Liu et al., 2020)	77.91	75.86	73.61	70.24
KGAT (reproduced)	78.20	75.98	73.66	70.06
KGAT + SR + weighting	**79.19**	**77.01**	**73.97**	**70.71**
Backbone: RoBERTa Large				
KGAT (Liu et al., 2020)	78.29	76.11	74.07	70.38
KGAT (reproduced)	80.19	78.03	75.40	72.04
KGAT + SR + weighting	**80.70**	**78.63**	**75.72**	**72.53**
Non-KGAT SOTA Methods				
Stammbach (Stammbach, 2021)	–	–	79.20	76.80
LisT5 (Jiang et al., 2021)	81.26	77.75	79.35	75.87
ProoFVer (Krishna et al., 2022)	80.74	79.07	79.47	76.82
BEVERS (DeHaven and Scott, 2023)	–	–	**80.24**	**77.70**

Table 3: Label accuracy (LA) and FEVER score (FS) on the development (Dev) and test sets. The bold values indicate the best performer in the group.

used with BERT Base.

Although the MLL loss explicitly has the additional penalty term for the complement sets, it does not account for the label heterogeneity as in the cross-entropy loss (see Section 2.2.1). Indeed, there is little difference in the results between CE and MLL, excluding the BERT Large backbone without weighting.

Comparison with SOTA models As KGAT with the proposed SR objective and class-balanced weighting showed consistent performance on the development set, we submit its predictions on the test set to the FEVER scoring site. Table 3 presents the results, along with those of the original KGAT and state-of-the-art (SOTA) FEVER models. The proposed methods (KGAT + SR + weighting) consistently outperform the original KGAT (using the standard CE loss) on the test set as well, regardless of the backbone architecture. These results suggest that the cross-entropy objective is not necessarily optimal for the FEVER task, and our approach offers a means of improvement.

The scores of KGAT models, including our proposed approach, are lower than those of the SOTA models (Stammbach, 2021; Jiang et al., 2021; Krishna et al., 2022; DeHaven and Scott, 2023). However, it should be noted that these models owe their better performance in part to the improved retrievers and backbones they use. Indeed, DeHaven and Scott (2023, Table 12) report an LA of 76.60 and an FS of 73.21 on the test set, when their BEVERS model is used in combination with the KGAT retriever and the RoBERTa Large backbone. These figures represent a notable regression from those presented in Table 3, consequently reducing the advantage over our model (with a test LA of 75.72, and a test FS of 72.53) to less than a 1-point.

4 Related Work

The FEVER shared tasks (Thorne et al., 2018, 2019; Aly et al., 2021a,b) have been the subject of extensive research. Most proposed approaches utilize Transformer-based models to embed claims and evidence (Tymoshenko and Moschitti, 2021; Jiang et al., 2021; Stammbach, 2021; DeHaven and Scott, 2023), whereas some researchers (Zhou et al., 2019; Liu et al., 2020) use graph-based methods to aggregate information from multiple pieces of evidence. None of these studies focus on the objective function to optimize, and most employ the standard cross-entropy objective.

Recently, DeHaven and Scott (2023) have used class weighting to mitigate class imbalance in the FEVER dataset, although the detailed weighting scheme is not reported.

In machine learning, Zhang (2004) analyzes various loss functions used for multiclass classification, including a general form of one-versus-rest (or one-versus-all) loss functions, which also have terms accounting for the complement set of the ground-truth class. Ishida et al. (2017) study complementary-label learning scenarios (Ishida et al., 2017; Yu et al., 2018; Ishida et al., 2019) extending Zhang's losses.

5 Conclusion

We introduced loss functions that take into account the heterogeneity of verdict classes in the FEVER task. In empirical evaluation, they consistently outperformed the standard cross-entropy loss.

In future work, we will evaluate the proposed loss functions in other fact verification tasks. We also plan to apply them to SOTA models for FEVER. As these models use the cross-entropy loss, our auxiliary loss terms are readily applicable.

Limitations

Our empirical evaluation was conducted in limited situations.

- Task (dataset): Although our approach proved effective in the FEVER task and dataset

(Thorne et al., 2018), whether it works equally well in other similar tasks and datasets remains unverified.

- Verdict predictor: The effectiveness of our approach was demonstrated only in combination with KGAT (Liu et al., 2020), a popular prediction model frequently used for benchmarking FEVER methods. Being model-agnostic, our loss functions need to be evaluated in combination with more recent models that optimize the cross-entropy loss.

Acknowledgments

We are grateful to anonymous reviewers for their constructive comments. This work is partially supported by JSPS Kakenhi Grant 21K17811 to YS.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021a. 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.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021b. FEVEROUS: Fact extraction and VERification over unstructured and structured information. In *Proceedings of the 35th Conference on Neural Information Processing Systems, Datasets and Benchmarks Track (Round 1)*.

Eric B. Baum and Frank Wilczek. 1988. Supervised learning of probability distributions by neural networks. In *Neural Information Processing Systems*. American Institute of Physics.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277.

Mitchell DeHaven and Stephen Scott. 2023. BEVERS: A general, simple, and performant framework for automatic fact verification. In *Proceedings of the Sixth Fact Extraction and VERification Workshop (FEVER)*, pages 58–65, Dubrovnik, Croatia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. 2017. Learning from complementary labels. In *Advances in Neural Information Processing Systems*.

Takashi Ishida, Gang Niu, Aditya Menon, and Masashi Sugiyama. 2019. Complementary-label learning for arbitrary losses and models. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2971–2980.

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.

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. ProoFVer: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030.

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 preprint 1907.11692 [cs.CL].

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12:153–157.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4334–4343.

Dominik Stammbach. 2021. Evidence selection as a token-level prediction task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 14–20, Dominican Republic. 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.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6, Hong Kong, China. Association for Computational Linguistics.

Kateryna Tymoshenko and Alessandro Moschitti. 2021. Strong and light baseline models for fact-checking joint inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4824–4830, Online. Association for Computational Linguistics.

Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. 2018. Learning with biased complementary labels. In *Proceedings of the European Conference on Computer Vision*, pages 68–83.

Tong Zhang. 2004. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. 2023. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10795–10816.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901, Florence, Italy. Association for Computational Linguistics.

A Summary of Objective Functions

In the following, we list the formulas for the objective functions used in our experiments.

Cross-entropy objective The cross-entropy objective presented in Eq. (1) is repeated here for convenience.

$$L_{\mathrm{CE}}(\mathbf{y},\mathbf{p}) = -\sum_{i=1}^{3} y_i \log p_i .$$

Its class-weighted version is:

$$L_{\mathrm{CE\ \ weighting}}(\mathbf{y},\mathbf{p}) = -\sum_{i=1}^{3} w_i y_i \log p_i .$$

MLL objective The MLL objective of Eq. (2) is:

$$\begin{aligned} L_{\mathrm{MLL}}(\mathbf{y},\mathbf{p}) &= L_{\mathrm{CE}}(\mathbf{y},\mathbf{p}) \quad \lambda R_{\mathrm{MLL}}(\mathbf{y},\mathbf{p}) \\ &= -\sum_{i=1}^{3} \left[y_i \log p_i \quad \lambda(1-y_i)\log(1-p_i) \right], \end{aligned}$$

and its weighted version is:

$$\begin{aligned} &L_{\mathrm{MLL\ \ weighting}}(\mathbf{y},\mathbf{p}) \\ &= -\sum_{i=1}^{3} w_i \left[y_i \log p_i \quad \lambda(1-y_i)\log(1-p_i) \right]. \end{aligned}$$

SRN objective The SRN objective L_{SRN}, originally presented in Eq. (7), is restated below, accompanied by its instantiation for individual gold classes:

$$\begin{aligned} L_{\mathrm{SRN}}(\mathbf{y},\mathbf{p}) &= L_{\mathrm{CE}}(\mathbf{y},\mathbf{p}) \quad \lambda R_{\mathrm{SRN}}(\mathbf{y},\mathbf{p}) \\ &= -\sum_{i=1}^{3} y_i \log p_i - \lambda \sum_{i=1}^{2} (1-y_i)\log(1-p_i) \\ &= \begin{cases} -\log p_1 - \log(1-p_2), & \text{if } y_1 = 1, \\ -\log p_2 - \log(1-p_1), & \text{if } y_2 = 1, \\ -\log p_3 - \log(1-p_1) & \\ \qquad\qquad - \log(1-p_2), & \text{if } y_3 = 1. \end{cases} \end{aligned}$$

With class weighting, the objective becomes Eq. (10), as shown in Section 2.2. The corresponding expressions for individual gold classes are as follows:

$$\begin{aligned} &L_{\mathrm{SRN\ \ weighting}}(\mathbf{y},\mathbf{p}) \\ &= \begin{cases} -w_1 \left[\log p_1 \quad \log(1-p_2)\right], & \text{if } y_1 = 1, \\ -w_2 \left[\log p_2 \quad \log(1-p_1)\right], & \text{if } y_2 = 1, \\ -w_3 \left[\log p_3 \quad \log(1-p_1)\right. & \\ \qquad\qquad \left.\log(1-p_2)\right], & \text{if } y_3 = 1. \end{cases} \end{aligned}$$

SR objective The objective L_{SR} is shown below:

$$\begin{aligned} L_{\mathrm{SR}}(\mathbf{y},\mathbf{p}) &= L_{\mathrm{CE}}(\mathbf{y},\mathbf{p}) \quad \lambda R_{\mathrm{SR}}(\mathbf{y},\mathbf{p}) \\ &= -\sum_{i=1}^{3} y_i \log p_i \\ &\qquad - \lambda(1-y_3)\sum_{i=1}^{2}(1-y_i)\log(1-p_i) \\ &= \begin{cases} -\log p_1 - \log(1-p_2), & \text{if } y_1 = 1, \\ -\log p_2 - \log(1-p_1), & \text{if } y_2 = 1, \\ -\log p_3, & \text{if } y_3 = 1. \end{cases} \end{aligned}$$

And the weighted version is:

$$L_{\mathrm{SR\ \ weighting}}(\mathbf{y},\mathbf{p})$$

$$= \begin{cases} -w_1 \left[\log p_1 \quad \log(1 - p_2)\right], & \text{if } y_1 = 1, \\ -w_2 \left[\log p_2 \quad \log(1 - p_1)\right], & \text{if } y_2 = 1, \\ -w_3 \log p_3, & \text{if } y_3 = 1. \end{cases}$$

B Additional Experimental Results

B.1 Confusion Matrices

To provide a comprehensive view of the compared prediction models, the confusion matrices of their predictions are presented in Tables 4–6. We observe that the sample weighting mitigates the imbalance bias in most cases. Specifically, weighting decreases the number of predictions for the majority class (SUP), for example, from 7497 to 7211 in the case of the BERT Base backbone; compare Table 4(a) and (b).

B.2 Effect of λ

We introduced in the MLL objective of Eq. (2) a hyperparameter λ to balance the primary and auxiliary terms in the objective.

To evaluate the efficacy of calibrating the λ parameter, we specifically examine the performance for fixed $\lambda = 1$ (i.e., direct application of original MLL loss), and that of λ tuned over the development set. Table 7 shows the results. We note that the scores of $\lambda = 1$ are considerably lower than those achieved when λ is optimized on the development set.

C License of the Assets

The FEVER 2018 dataset[3] is licensed under the CC BY-SA 3.0. The KGAT implementation[4] is licensed under the MIT License.

[3] https://fever.ai/dataset/fever.html
[4] https://github.com/thunlp/KernelGAT

		Prediction		
		SUP	REF	NEI
Gold	SUP	5976	222	468
	REF	470	5153	1043
	NEI	1051	1184	4431
Total		7497	6559	5942

(a) Loss = CE, Weighting = no (FS=75.75, LA=77.81)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5862	214	590
	REF	427	4906	1333
	NEI	922	897	4847
Total		7211	6017	6770

(b) Loss = CE, Weighting = yes (FS=76.02, LA=78.08)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5976	201	489
	REF	510	4981	1175
	NEI	1066	991	4609
Total		7552	6173	6273

(c) Loss = MLL, Weighting = no (FS=75.65, LA=77.84)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5785	303	578
	REF	372	5098	1196
	NEI	845	1079	4742
Total		7002	6480	6516

(d) Loss = MLL, Weighting = yes (FS=76.06, LA=78.13)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5919	196	551
	REF	455	4876	1335
	NEI	1001	894	4771
Total		7375	5966	6657

(e) Loss = SRN, Weighting = no (FS=75.70, LA=77.84)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5766	239	661
	REF	444	4958	1264
	NEI	864	962	4840
Total		7074	6159	6765

(f) Loss = SRN, Weighting = yes (FS=75.79, LA=77.83)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5948	221	497
	REF	461	4969	1236
	NEI	1014	939	4713
Total		7423	6129	6446

(g) Loss = SR, Weighting = no (FS=75.87, LA=78.16)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5979	228	459
	REF	457	5031	1178
	NEI	1080	939	4647
Total		7516	6198	6284

(h) Loss = SR, Weighting = yes (FS=76.06, LA=78.29)

Table 4: Confusion matrices on the development set, with the BERT Base backbone. The “Total” row shows the number of times each class is predicted.

		Prediction		
		SUP	REF	NEI
Gold	SUP	5985	222	459
	REF	436	5061	1169
	NEI	1032	1042	4592
Total		7453	6325	6220

(a) Loss = CE, Weighting = no (FS=75.98, LA=78.20)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5817	238	611
	REF	349	5171	1146
	NEI	854	1032	4780
Total		7020	6441	6537

(b) Loss = CE, Weighting = yes (FS=76.74, LA=78.85)

		Prediction		
		SUP	REF	NEI
Gold	SUP	6011	188	467
	REF	437	5068	1161
	NEI	1019	940	4707
Total		7467	6196	6335

(c) Loss = MLL, Weighting = no (FS=76.78, LA=78.94)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5858	258	550
	REF	359	5214	1093
	NEI	858	1112	4696
Total		7075	6584	6339

(d) Loss = MLL, Weighting = yes (FS=76.74, LA=78.85)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5942	214	510
	REF	406	5076	1184
	NEI	922	1028	4716
Total		7270	6318	6410

(e) Loss = SRN, Weighting = no (FS=76.57, LA=78.68)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5806	246	614
	REF	323	5148	1195
	NEI	852	1004	4810
Total		6981	6398	6619

(f) Loss = SRN, Weighting = yes (FS=76.71, LA=78.83)

		Prediction		
		SUP	REF	NEI
Gold	SUP	6024	165	477
	REF	411	4989	1266
	NEI	1007	869	4790
Total		7442	6023	6533

(g) Loss = SR, Weighting = no (FS=76.86, LA=79.02)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5938	187	541
	REF	397	5087	1182
	NEI	884	971	4811
Total		7219	6245	6534

(h) Loss = SR, Weighting = yes (FS=77.01, LA=79.19)

Table 5: Confusion matrices on the development set, with the BERT Large backbone.

		Prediction		
		SUP	REF	NEI
Gold	SUP	6073	153	440
	REF	357	5127	1182
	NEI	964	865	4837
Total		7394	6145	6459

(a) Loss = CE, Weighting = no (FS=78.03, LA=80.19)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5783	220	663
	REF	238	5291	1137
	NEI	693	938	5035
Total		6714	6449	6835

(b) Loss = CE, Weighting = yes (FS=78.54, LA=80.55)

		Prediction		
		SUP	REF	NEI
Gold	SUP	6032	148	486
	REF	321	5092	1253
	NEI	913	878	4875
Total		7266	6118	6614

(c) Loss = MLL, Weighting = no (FS=77.88, LA=80.00)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5995	159	512
	REF	299	5151	1216
	NEI	826	864	4976
Total		7120	6174	6704

(d) Loss = MLL, Weighting = yes (FS=78.55, LA=80.62)

		Prediction		
		SUP	REF	NEI
Gold	SUP	6117	129	420
	REF	361	4996	1309
	NEI	962	771	4933
Total		7440	5896	6662

(e) Loss = SRN, Weighting = no (FS=78.18 LA=80.24)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5913	227	526
	REF	275	5410	981
	NEI	780	1064	4822
Total		6968	6701	6329

(f) Loss = SRN, Weighting = yes (FS=78.56, LA=80.73)

		Prediction		
		SUP	REF	NEI
Gold	SUP	6072	162	432
	REF	314	5239	1113
	NEI	915	981	4770
Total		7301	6382	6315

(g) Loss = SR, Weighting = no (FS=78.19, LA=80.41)

		Prediction		
		SUP	REF	NEI
Gold	SUP	5901	213	552
	REF	237	5238	1191
	NEI	766	901	4999
Total		6904	6352	6742

(h) Loss = SR, Weighting = yes (FS=78.63, LA=80.70)

Table 6: Confusion matrices on the development set, with the RoBERTa Large backbone.

Backbone	Loss	Weighting	LA	FS
BERT Base	MLL (λ = 0.125)	yes (β = 0.999999)	**78.13**	**76.06**
	MLL (λ = 1)	yes (β = 0.99999)	77.96	75.91
BERT Large	MLL (λ = 0.03125)	yes (β = 0.999999)	**78.85**	**76.74**
	MLL (λ = 1)	yes (β = 0.999999)	78.68	76.56
RoBERTa Large	MLL (λ = 0.0625)	yes (β = 0.999999)	**80.62**	**78.55**
	MLL (λ = 1)	yes (β = 0.99999)	80.05	77.97

Table 7: Effect of tuning λ in the MLL objective.