Simple Compounded-Label Training for Fact Extraction and Verification

Yixin Nie[*] **Lisa Bauer**[*] **Mohit Bansal**
UNC Chapel Hill
{yixin1, lbauer6, mbansal}@cs.unc.edu

Abstract

Automatic fact checking is an important task motivated by the need for detecting and preventing the spread of misinformation across the web. The recently released FEVER challenge provides a benchmark task that assesses systems' capability for both the retrieval of required evidence and the identification of authentic claims. Previous approaches share a similar pipeline training paradigm that decomposes the task into three subtasks, with each component built and trained separately. Although achieving acceptable scores, these methods induce difficulty for practical application development due to unnecessary complexity and expensive computation. In this paper, we explore the potential of simplifying the system design and reducing training computation by proposing a joint training setup in which a single sequence matching model is trained with compounded labels that give supervision for both sentence selection and claim verification subtasks, eliminating the duplicate computation that occurs when models are designed and trained separately. Empirical results on FEVER indicate that our method: (1) outperforms the typical multi-task learning approach, and (2) gets comparable results to top performing systems with a much simpler training setup and less training computation (in terms of the amount of data consumed and the number of model parameters), facilitating future works on the automatic fact checking task and its practical usage.

1 Introduction

The increasing concern with the spread of misinformation has motivated research regarding automatic fact checking datasets and systems (Pomerleau and Rao, 2017; Hanselowski et al., 2018a; Bast et al., 2017; Pérez-Rosas et al., 2018; Zhou et al., 2019; Vlachos and Riedel, 2014; Wang, 2017; Shu et al., 2019a,b). The Fact Extraction and VERification (FEVER) dataset (Thorne et al., 2018a) is the most recent large-scale dataset that enables the development of data-driven neural approaches to the automatic fact checking task. Additionally, the FEVER Shared Task (Thorne et al., 2018b) introduced a benchmark, the first of this kind, that is capable of evaluating both evidence retrieval and claim verification.

Several top-ranked approaches on FEVER (Nie et al., 2019a; Yoneda et al., 2018; Hanselowski et al., 2018b) decompose the task into 3 subtasks: document retrieval, sentence selection, and claim verification, and follow a similar pipeline training setup where sub-components are developed and trained sequentially. Although achieving higher scores on benchmarks, pipeline training is time-consuming and imposes difficulty for fast application development since downstream training relies on data provided by a fully-converged upstream component. The impossibility of parallelization also causes data-inefficiency as training the same input sentence for both sentence selection and claim verification requires twice the computation, whereas humans can learn the task of sentence selection and claim verification jointly.

In this work, we simplify the training procedure and increase training efficiency for sentence selection and claim verification by merging redundant components and computation that exist when training the two tasks separately. We propose a joint training setup in which sentence selection and claim verification are tackled by a single neural sequence matching model. This model is trained with a *compounded label space* in which for a given claim, an input sentence that is labeled as "NON-SELECT" for sentence selection module training will also be labeled as "NOTENOUGHINFO" for claim verification module training. Similarly, input evidence that is labeled as "SUPPORTS" or

Our code will be publicly available on our webpage.

[*] Equal contribution

Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER), pages 1–7
July 9, 2020. ©2020 Association for Computational Linguistics

"REFUTES" for claim verification module training will also be labeled as "SELECT" for sentence selection module training.

To validate our new setup, we compare with the previous pipeline setup and a multi-task learning setup which trains the two tasks alternately. Fig. 1 illustrates differences among these three setups.

Results indicate that: our method (1) outperforms the multi-task learning setup, and (2) yields comparable results with a top performing pipeline-trained system while consuming less than half the number of data points, reducing the parameter size by one-third, and converging to a functional state much faster than the pipeline-trained system. We argue that the aforementioned design simplification and training acceleration are valuable especially during time-sensitive application development.

2 Related Work

2.1 Previous FEVER Systems

Many of the top performing FEVER 1.0 systems, all achieving greater than 60% FEVER score on the respective leaderboard (Nie et al., 2019a; Yoneda et al., 2018; Hanselowski et al., 2018b), share the same pipeline training schema in which document retrieval, sentence selection, and claim verification are all trained separately.

While Nie et al. (2019a) proposed formalizing sentence selection and claim verification as a similar problem, sentence selection and claim verification are still trained separately on the task, which contrasts with our setup. Additionally, Yin and Roth (2018) proposed a hierarchical neural model to tackle both sentence selection and claim verification at the same time, but did not induce computational savings as in our setup.

2.2 Information Retrieval

Neural networks have been successfully applied to information retrieval tasks in Natural Language Processing (Huang et al., 2013; Guo et al., 2016; Mitra et al., 2017; Dehghani et al., 2017; Qi et al., 2019; Nie et al., 2019b) with a focus on relevant retrieval. Information retrieval is generally a relevance-matching task whereas claim verification is a more semantics-intensive task. We consider using a single semantics-focused model to conduct both sentence retrieval and claim verification.

2.3 Natural Language Inference

Natural Language Inference (NLI) requires a system to classify the logical relationship between two sentences in which one is the premise and one is the hypothesis. This classifier decides whether the relationship is entailment, contradiction, or neutral. Several large-scale datasets have been created for this purpose, including the Stanford Natural Language Inference Corpus (Bowman et al., 2015) and the Multi-Genre Natural Language Inference Corpus (Williams et al., 2018). This task can be formalized as a semantic sequence matching task, which bears resemblance to both the sentence retrieval and claim verification tasks.

2.4 Multi-Task Learning

Multi-task learning (MTL) (Caruana, 1997) has been successfully used to merge Natural Language Processing tasks (Luong et al., 2016; Hashimoto et al., 2017; Dong et al., 2015) for improved performance. Parameter sharing, in particular sharing of certain structures such as label spaces, has been used widely in several NLP tasks for this purpose (Liu et al., 2017; Søgaard and Goldberg, 2016). Zhao et al. (2018) used a multi-task learning setup for FEVER that shared certain layers between sentence selection and claim verification modules. Augenstein et al. (2018) used shared label spaces in MTL for sequence classification. Following this work, Augenstein et al. (2019) used shared label spaces for automatic fact checking. However, the labels involved in this work were limited to claim verification labels only, and did not incorporate sentence selection as we do in this paper.

2.5 Fake News Detection

In addition to the FEVER shared task, other recent work in fake news detection has focused on several aspects of data collection and statement verification. Shu et al. (2019b) looked into the role of social context in fake news detection. Additionally, Shu et al. (2019a) also explored creating explainable fake news detection.

3 Model

3.1 Sequence Matching Model

Sentence selection and claim verification can be easily structured as the same sequence matching problem in which the input is a pair of textual sequences and the output is a semantic relationship label for the pair. Nie et al. (2019a) proposed using

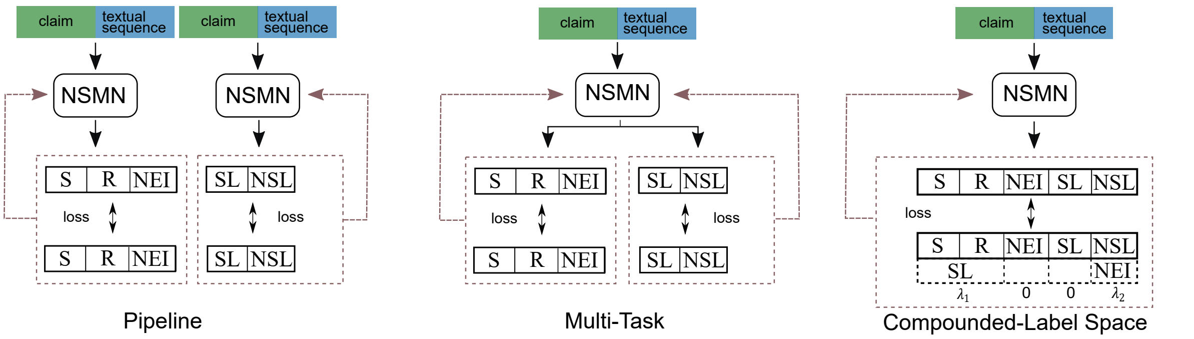

Figure 1: Different training setups. In the pipeline setup, sentence selection and claim verification models are trained separately. In the multi-task setup, the two tasks are treated separately, but use a single model. In the compounded-label training setup, the training is simplified to a single task by mixing the data of the two tasks and allowing controlled supervision between the two tasks. S, R, NEI, SL, and NSL represent "SUPPORTS", "REFUTES", "NOTENOUGHINFO", "SELECT", and "NON-SELECT", respectively.

the same architecture, the neural semantic matching network (NSMN), on the two tasks and showed it was effective on both. Thus, we use the same NSMN model with a modified output layer in our experiments.

3.2 Neural Semantic Matching Network (NSMN)

For convenience, we give a description similar to the original paper (Nie et al., 2019a) about the model below.

Encoding Layer:

$$\bar{\mathbf{U}} = \text{BiLSTM}_e(\mathbf{U}) \in \mathbb{R}^{d_1 \times n} \tag{1}$$

$$\bar{\mathbf{H}} = \text{BiLSTM}_e(\mathbf{H}) \in \mathbb{R}^{d_1 \times m} \tag{2}$$

where $\mathbf{U} \in \mathbb{R}^{d_0 \times n}$ and $\mathbf{H} \in \mathbb{R}^{d_0 \times m}$ are the two input sequences, d_0 and d_1 are input and output dimensions, and n and m are lengths of the two sequences.

Alignment Layer:

$$\mathbf{A} = \bar{\mathbf{U}}^\top \bar{\mathbf{H}} \in \mathbb{R}^{n \times m} \tag{3}$$

where an element in $\mathbf{A}_{[i,j]}$ indicates the alignment score between i-th token in $\mathbf{U}$ and j-th token in $\mathbf{H}$. Aligned sequences are computed as:

$$\tilde{\mathbf{U}} = \bar{\mathbf{H}} \cdot \text{Softmax}_{\text{col}}(\mathbf{A}^\top) \in \mathbb{R}^{d_1 \times n} \tag{4}$$

$$\tilde{\mathbf{H}} = \bar{\mathbf{U}} \cdot \text{Softmax}_{\text{col}}(\mathbf{A}) \in \mathbb{R}^{d_1 \times m} \tag{5}$$

where $\text{Softmax}_{\text{col}}$ is column-wise softmax, $\tilde{\mathbf{U}}$ is the aligned representation from $\bar{\mathbf{H}}$ to $\bar{\mathbf{U}}$ and vice versa for $\tilde{\mathbf{H}}$. The aligned and encoded representations are combined as:

$$\mathbf{F} = f([\bar{\mathbf{U}}, \tilde{\mathbf{U}}, \bar{\mathbf{U}} - \tilde{\mathbf{U}}, \bar{\mathbf{U}} \circ \tilde{\mathbf{U}}]) \in \mathbb{R}^{d_2 \times n} \tag{6}$$

$$\mathbf{G} = f([\bar{\mathbf{H}}, \tilde{\mathbf{H}}, \bar{\mathbf{H}} - \tilde{\mathbf{H}}, \bar{\mathbf{H}} \circ \tilde{\mathbf{H}}]) \in \mathbb{R}^{d_2 \times m} \tag{7}$$

where f is one fully-connected layer with a rectifier as an activation function and $\circ$ denotes element-wise multiplication.

Matching Layer:

$$\mathbf{R} = \text{BiLSTM}_m([\mathbf{F}, \mathbf{U}^*]) \in \mathbb{R}^{d_3 \times n} \tag{8}$$

$$\mathbf{S} = \text{BiLSTM}_m([\mathbf{G}, \mathbf{H}^*]) \in \mathbb{R}^{d_3 \times m} \tag{9}$$

where $\mathbf{U}^*$ and $\mathbf{H}^*$ are sub-channels of the input $\mathbf{U}$ and $\mathbf{H}$ without GloVe, provided to the matching layer via a shortcut connection.

Output Layer:

$$\mathbf{r} = \text{Maxpool}_{\text{row}}(\mathbf{R}) \in \mathbb{R}^{d_3} \tag{10}$$

$$\mathbf{s} = \text{Maxpool}_{\text{row}}(\mathbf{S}) \in \mathbb{R}^{d_3} \tag{11}$$

$$h(\mathbf{r}, \mathbf{s}, |\mathbf{r} - \mathbf{s}|, \mathbf{r} \circ \mathbf{s}) = \mathbf{m} \tag{12}$$

where function h denotes two fully-connected layers with a rectifier being applied on the output of the first layer.

3.3 Compounded-Label Output Layer

We propose the following compounded-label output layer for simpler, more efficient training. Given the input pair x_i, the NSMN model is:

$$\mathbf{m} = \text{NSMN}(x_i) \tag{13}$$

where $\mathbf{m} \in \mathbb{R}^4$ is the output vector of NSMN in which the first three elements correspond to claim verification and the last element to sentence selection. Then, the probabilities are calculated as:

$$\mathbf{y}_{cv} = \text{softmax}(\mathbf{m}_{[0:3]}) \tag{14}$$

$$y_{ss} = \text{sigmoid}(m_3) \tag{15}$$

where $\mathbf{m}_{[0:3]}$ denotes the first three elements of $\mathbf{m}$ and $\mathbf{y}_{cv} \in \mathbb{R}^3$ denotes the probability of predicting the relation between the input and claim as "SUPPORTS", "REFUTES", or

"NotEnoughInfo", while m_3 denotes the fourth element of $\mathbf{m}$ and $y_{ss} \in \mathbb{R}$ indicates the probability of choosing the input as evidence for the claim. This allows us to transfer the model's outputs to predictions in a compact way.

3.4 Compounded-Label Training

In order to simplify the training procedure and increase data efficiency, we introduce compounded-label training. Consider the model output vector:

$$\hat{\mathbf{y}}_i = \left[\begin{array}{c} \mathbf{y}_{cv} \\ \hline y_{ss} \\ 1 - y_{ss} \end{array}\right] \tag{16}$$

where $\hat{\mathbf{y}}_i \in \mathbb{R}^5$ is the concatenation of $\mathbf{y}_{cv}$ and $[y_{ss}, 1 - y_{ss}]^\top$. To optimize the model, we use the entropy objective function:

$$\mathcal{J} = -\mathbf{y}_i \cdot \log(\hat{\mathbf{y}}_i) \tag{17}$$

In a typical classification setup, the ground truth label embedding y_i is a one-hot column vector chosen from an identity matrix, where the dimension equals the total number of categories. However, our compounded-label embedding is structured as the matrix with some supervision provided in the zero-area of one-hot embeddings shown below:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \lambda_2 \\ \lambda_1 & \lambda_1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

The first 3 columns are label embeddings for "Supports", "Refutes", and "NotEnoughInfo" in verification and the last 2 columns are the label embeddings for "Select" and "Non-select" in sentence selection, resp. Thus, for a given claim, "Supports" and "Refutes" evidence will also give supervision as positive examples to sentence selection weighted by λ_1 and "Non-select" sentences will also give supervision as "NotEnoughInfo" evidence to claim verification weighted by λ_2.

4 Experimental Setup

We focused on comparing the following five NSMN[1] training setups for sentence selection and claim verification. We obtain upstream document retrieval data using the method in Nie et al. (2019a). Training details are in the appendix.

	Pip.	Mul.	Mix.	Cmp.
Shared Parameters	✗	✓	✓	✓
Mix. in Same Batch	✗	✗	✓	✓
Supv. for Other Task	✗	✗	✗	✓

Table 1: Properties of different training setups. "**Pip.**", "**Mtl.**", "**Mix.**", "**Cmp.**" stand for pipeline, multi-task learning, direct mixing, and compounded-label training setup, respectively. 'Supv.'=Supervision.

Model	FEVER Score	Rec.	# Param	Data
D.M.	57.92	85.3	18.2M	11.5M
MTL.	62.25	85.3	18.2M	14.4M
Rdc-Pip.	61.82	83.7	18.2M	11.4M
C.L.	64.68	86.6	18.2M	3.52M
Pip.	65.37	86.8	27.6M	9.6M

Table 2: Final performance, evidence recall, model size, and data consumption (until convergence) for all 5 setups. We measure data consumption as the amount of data the model used for parameter updating, e.g., 10K updates w/ batch size 32 consumes 320K data. 'D.M.'=direct mixing, 'C.L.'=compounded-label, 'MTL.'=multi-task learning, 'Rdc-Pip.'=pipeline w/ reduced size, 'Pip.'=pipeline (Nie et al., 2019a).

Pipeline: We train separate sentence selection and claim verification models as in Nie et al. (2019a).

Multi-task Learning: We follow the neural multi-task learning setup called alternate training (Dong et al., 2015; Luong et al., 2016; Hashimoto et al., 2017), where each batch contains examples from a single task only. We build a single NSMN model for both selection and verification and alternatively optimize the two tasks.

Direct Mixing: We simply blend the input examples of the two tasks into the same batch, providing additional simplicity over our multi-task learning setup in which batches need to be task-exclusive.

Compounded-Label Training: We also blend the inputs of the two tasks, but counter to direct mixing, we use the compounded-label embedding described in Sec. 3 for optimization and downsample the input examples to reduce training time.

Reduced Pipeline: This is the same pipeline setup as described above, except that we reduce the model sizes for both sentence selection and verification such that the total model size is equal to all other setups that use only a single joint model. This experiment gives a fair comparison between each of the setups by canceling out the parameter-size variance. Table 1 shows a comparison of the first four different setups.

[1] We remove the external WordNet features from NSMN for simplicity and speed.

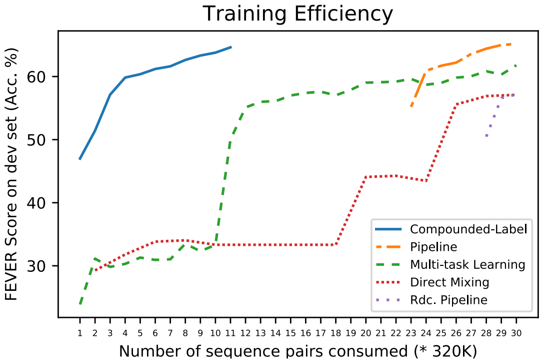

Figure 2: Model performance for different setups with respect to number of sequence pairs consumed. We only show performance until the consumption of the first 30×320K data points.

5 Results and Analysis

FEVER Score Performance: We observe from Table 2 that compounded-label training outperforms[2] both the multitask learning and direct mixing setups. We speculate that the performance gap is due to the fact that in the multi-task and direct mixing training setups, the same model is trained by separated and different supervisions of two tasks, resulting in oscillation and making it difficult to reach a better global minimum. However, in the compounded-label setup, training the model on one task always gives a subtly-controlled supervision on the other task. This not only applies natural regularization on the targeted task itself, but also pushes the model towards a better state for both tasks.

Next, we also show that the compounded-label setup achieves a higher FEVER score than the reduced-pipeline setup (3rd row in Table 2), indicating its ability to model the two tasks jointly in a more compact and parameter-efficient way. Although the full pipeline setup gives a slightly higher FEVER score, the compounded-label setup has the advantage of reducing parameter size by one-third, requiring less than half the training computation, and improving the training efficiency (elaborated on in the following subsection). Finally, we also compare recall scores, since this is most related to the FEVER score, as validated by Nie et al. (2019a).

Efficiency: In Fig. 2, we show the training efficiency of different approaches by tracking performance with the number of data points consumed.[3] Parameter update settings are equal across all experiments and thus show an accurate depiction of the speedup independent of batch size, etc. For fair comparison, there is no FEVER score for the first 22 × 320K data points in the pipeline setup since these data points are consumed in the separate upstream sentence selection training. The compounded-label training setup exhibits a more stable training curve than the other setups during initial training, and reaches a 60%+ FEVER score after seeing only 1,280K data points. This indicates that the compounded-label setting allows the model to quickly reach a stable and functional state. This is valuable for online learning on streaming data, where the model is trained with real-time human feedback. On the contrary, the performance of the multi-task learning and direct mixing setups fluctuates at a low level during initial training stages, which shows that optimization oscillation makes training difficult in these setups.

Model	FEVER	LA	F1
Pipeline	**62.69**	66.20	53.71
Compounded-Label	61.65	**66.21**	50.28

Table 3: Performance of systems on blind test results.

Blind Test Results: In Table 3 we compare the two setups on the blind test set. Compounded Label achieved 61.65% FEVER score and 66.21% label score (LA) while the pipeline setup got 62.69% and 66.20% for FEVER score and LA, respectively. Since the upper bound is dependent on document retrieval quality, we report the upper bound of these scores as 92.42% following Nie et al. (2019a). Our method was able to yield results comparable to the pipeline model on FEVER score and even higher results on label score, with simpler design, faster convergence and only two-thirds the number of parameters.

6 Conclusion

We present a simple compounded-label setup for jointly training sentence selection and claim verification. This setup provides higher training efficiency and lower parameter size while still achieving comparable results to the pipeline approach.

[2] In Table 2, the improvements of compounded-label over the first three entries are significant with $p < 10^{-5}$ while the improvement of full pipeline over compounded-label is significant with $p < 0.05$. Stat. significance was computed on bootstrap test with 100K iterations (Noreen, 1989; Efron and Tibshirani, 1994).

[3] We measure the training efficiency based on the size of data consumed until convergence rather than training time or the full training size because it gives a fair measurement about how fast the model can reach a fully-functional state independent of computational resources and platforms.

Acknowledgments

We thank the reviewers for their helpful comments. This work was supported by DARPA MCS Grant N66001-19-2-4031, and awards from Google and Facebook. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the funding agency.

References

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims. In *EMNLP*.

Isabelle Augenstein, Sebastian Ruder, and Anders Søgaard. 2018. Multi-task learning of pairwise sequence classification tasks over disparate label spaces. In *NAACL-HLT*.

Hannah Bast, Björn Buchhold, and Elmar Haussmann. 2017. Overview of the triple scoring task at the wsdm cup 2017. *WSDM Cup*.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *SIGIR*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *ACL*.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *CIKM*.

Andreas Hanselowski, Avinesh P.V.S., Benjamin Schiller, Felix Caspelherr, Debanjan * Chaudhuri, Christian M. Meyer, and Iryna Gurevych. 2018a. A retrospective analysis of the fake news challenge stance-detection task. In *COLING*.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018b. Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*.

Kazuma Hashimoto, Yoshimasa Tsuruoka, Richard Socher, et al. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *EMNLP*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. *ACL*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. *ICLR*.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *WWW*.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019a. Combining fact extraction and verification with neural semantic matching networks. *AAAI*.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019b. Revealing the importance of semantic retrieval for machine reading at scale. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Eric W Noreen. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *COLING*.

Pomerleau and Rao. 2017. Fake news challenge.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D Manning. 2019. Answering complex open-domain questions through iterative query generation. *EMNLP*.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019a. defend: Explainable fake news detection.

Kai Shu, Suhang Wang, and Huan Liu. 2019b. Beyond news contents: The role of social context for fake news detection. ACM.

Anders Søgaard and Yoav Goldberg. 2016. Deep multitask learning with low level tasks supervised at lower layers. In *ACL*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and verification (fever) shared task. *arXiv preprint arXiv:1811.10971*.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *ACL LACSS Workshop*.

William Yang Wang. 2017. ”liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *ACL*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*.

Wenpeng Yin and Dan Roth. 2018. Twowingos: A two-wing optimization strategy for evidential claim verification. In *EMNLP*.

Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Ucl machine reading group: Four factor framework for fact finding (hexaf). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*.

Shuai Zhao, Bo Cheng, Hao Yang, et al. 2018. An end-to-end multi-task learning model for fact checking. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*.

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. *ACL*.

A Appendix

A.1 NSMN Output Layer Modifications

The dimension of final NSMN output vector can be customized depending on the downstream task. In the pipeline setting, multi-task learning setting, and direct mixing setting, $\mathbf{m} = \langle m^+, m^- \rangle$ for sentence selection, where $m^+ \in \mathbb{R}$ is a scalar value indicating the score for selecting the current sentence as evidence and m^- gives the score for discarding it. For claim verification, $\mathbf{m} = \langle m_s, m_r, m_n \rangle$, where the elements of the vector denote the score for predicting the three labels, namely SUPPORTS, REFUTES, and NEI, respectively. However, in the compounded-label setting, $\mathbf{m} \in \mathbb{R}^4$ and the model is optimized with a compact label embedding described in the paper.

A.2 Training Details

This section includes the training details for sentence selection and verification. We use the pageview method in Nie et al. (2019a) to obtain the same upstream document retrieval data for all of our four setups.

Pipeline: In the pipeline and the reduced-size-pipeline setup, we use exactly the same training setup as in Nie et al. (2019a) for sentence selection and claim verification.

Multi-task Learning: In this setup, we choose batch as 64 and use Adam optimizer with default initial parameters. The mixing ratio for sentence selection and claim verification is set to 1 thus the two tasks are both trained alternately every two batches. As in Nie et al. (2019a), we downsample the training data for the sentence selection task at the beginning of each epoch.

Data Mixing: We use a batch size of 64 and Adam optimizer with default settings. As our two subtasks contain different amounts of training data, we use the data size ratio as the task mixing ratio within each batch. We guarantee that each label is present at least once in each mini-batch.

Compounded-Label: We use a batch size of 32 and Adam optimizer with default settings. We downsample the negative examples for sentence selection with the probability of p (this is done at the beginning of every epoch) and randomly mix and shuffle the training data for both sentence selection and claim verification into one input set and train the single model with compounded-label as described in the paper. p is set to be 0.1 at the first epoch and 0.025 otherwise. λ_1 and λ_2 are set to be 1 and 0.5 respectively.

Hyper-parameter Selection: In the experiments for multi-task learning, data mixing and compounded-label settings, the batch size is chosen from either 64 or 32 by optimizing final FEVER Score.[4] In multi-task learning, the mixing ratio of sentence selection to claim verification is tuned from $\{1, 2\}$. For the compounded-label setting, λ_1 and λ_2 are tuned from $\{1, 0.9\}$ and $\{0.45, 0.5\}$ respectively based on the intuition that supporting and refuting sentences can be also treated as positive evidence examples with high confidence while partially relevant sentences that cannot verify the claim can be treated as weakly related evidence.

[4] We observed a failure of convergence when we choose batch size as 32 in multi-task learning settings.