Weakly- and Semi-supervised Evidence Extraction

Danish Pruthi, Bhuwan Dhingra, Graham Neubig, Zachary C. Lipton
Carnegie Mellon University
Pittsburgh, USA
{ddanish, bdhingra, gneubig, zlipton}@cs.cmu.edu

Abstract

For many prediction tasks, stakeholders desire not only predictions but also supporting evidence that a human can use to verify its correctness. However, in practice, evidence annotations may only be available for a minority of training examples (if available at all). In this paper, we propose new methods to combine few evidence annotations (strong semi-supervision) with abundant document-level labels (weak supervision) for the task of evidence extraction. Evaluating on two classification tasks that feature evidence annotations, we find that our methods outperform baselines adapted from the interpretability literature to our task. Our approach yields gains with as few as hundred evidence annotations.[1]

1 Introduction

Despite the success of deep learning for countless prediction tasks, practitioners often desire that these models not only be accurate but also provide *interpretations* or *explanations* (Caruana et al., 2015; Weld and Bansal, 2019). Unfortunately, these terms lack precise meaning, and across papers, such explanations purport to address a wide spectrum of desiderata, and it seems unlikely any one method could address them all (Lipton, 2018). In both computer vision (Ribeiro et al., 2016; Simonyan et al., 2013) and natural language processing (Lei et al., 2016; Lehman et al., 2019), proposed explanation methods often take the form of highlighting salient features of the input. These so-called *local explanations* are intended to highlight features that elucidate "the reasons behind predictions" (Ribeiro et al., 2016). However, this characterization of the problem remains under-specified.

In this paper, we instead focus on supplementing predictions with *evidence*, which we define as information that gives users the ability to quickly verify the correctness of predictions. Fortunately, for many problems, a localized portion of the input is sufficient to validate the predicted label. In a large image, a small patch of an image containing a hamster may be sufficient to render the "hamster" label applicable. Similarly, in a long clinical note, a single sentence may suffice to confirm a predicted diagnosis. This ability to verify results engenders trust among users, and increases adoption of the machine learning systems (Dzindolet et al., 2003; Herlocker et al., 2000; Ribeiro et al., 2016). In Table 1, we outline the characteristic differences between local explanations and evidence.

Thus motivated, we cast our problem as learning to extract evidence using both strong and weak supervision. The former takes the form of explicit, but scarce, manual annotations of evidence segments, whereas the latter is provided by documents and their class labels which we assume are relatively abundant.[2] In the extreme case where evidence annotations are available for all examples, our task reduces to a standard multitask learning problem. In the opposite extreme, where only weak supervision is available, we find ourselves back in the under-specified realm addressed by local explanations. While predictive tokens may be extracted using only weak supervision, evidence extraction requires some amount of strong supervision.

We draw inspiration from Zaidan and Eisner (2008), who study the reverse problem—how to leverage marked evidence spans to improve classification performance. We optimize the joint likelihood of class labels and evidence spans, given the input examples. We factorize our objective such that we first *classify, and then extract* the

[1] Code and datasets to reproduce our work are available at: https://github.com/danishpruthi/evidence-extraction.

[2] While the task formulation is broadly applicable, we limit to text classification tasks for the scope of this work.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3965–3970
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

	Explanations	**Evidence**
Objective	Elucidate “the reasons behind predictions”.	Enable users to quickly verify the predictions.
Evaluation	Explanations are specific to the model. No ground truth explanations to compare against.	Evidence is a characteristic of the task. Can be compared against human-labeled evidence.
Example	A horror movie that lacks cohesion.	A horror movie that lacks cohesion.

Table 1: Distinguishing local explanations from evidence snippets. In the illustrative example, the token horror is predictive of the negative sentiment as horror movies tend to get poorer ratings than movies from other genres (Kaushik et al., 2019), however, no expert would mark it to be the evidence justifying the negative review.

evidence. For classification, we use BERT (Devlin et al., 2019). The extraction task (a sequence tagging problem) is modeled using a linear-chain CRF (Lafferty et al., 2001). The CRF uses representations and attention scores from BERT as emission features, allowing the two tasks (i.e. classification and extraction) to benefit from shared parameters. Further, the evidence extraction module is conditioned on the class label, enabling the CRF to output different evidence spans tailored to each class label. This is illustrated in Table 2.

For baselines, we repurpose input attribution methods from the interpretability literature. Many approaches in this category first *extract, and then classify* (Lei et al., 2016; Lehman et al., 2019; Jain et al., 2020; Paranjape et al., 2020). Across two text sequence classification and evidence extraction tasks, we find our methods to outperform baselines. Encouragingly, we observe gains by using our approach with as few as 100 evidence annotations.

2 Related Work

We briefly discuss methods from the interpretability literature that aim to identify salient features of the input. Lei et al. (2016) propose an approach wherein a generator first extracts a subset of the text from the original input, which is further fed to an encoder that classifies the input by using only the extracted subset. The generator and encoder are trained end-to-end via REINFORCE-style optimization (Williams, 1992). However, follow-up work discovered the end-to-end training to be quite unstable with high variance in results (Bastings et al., 2019; Paranjape et al., 2020). Consequently, other approaches adopted the core idea of *extract, and then classify* in different forms: Lehman et al. (2019) decouple the extraction and prediction modules and train them individually with intermediate supervision; Jain et al. (2020) use heuristics, like attention scores, for extraction; and lastly, Paranjape et al. (2020) extract subsets that have high mutual information with the output variable and low mutual information with the input variable.

3 Extracting Evidence

Formally, let the training data consist of n points $\{(\mathbf{x_1}, y_1) ... (\mathbf{x_n}, y_n)\}$, where $\mathbf{x_i}$ is a document, y_i is the associated label. We assume that for m points ($m \ll n$) we also have evidence annotations $\mathbf{e_i}$, a binary vector such that $e_{ij} = 1$ if token x_{ij} is a part of the evidence, and 0 otherwise. The conditional likelihood of the output labels and evidence given the documents can be written as:

$$L = \prod_{i=1}^{n} p(y_i, \mathbf{e_i} | \mathbf{x_i})$$

We can factorize this likelihood in two ways. First,

$$\begin{aligned} L &= \prod_{i=1}^{n} p(\mathbf{e_i}|\mathbf{x_i}) \; p(y_i|\mathbf{x_i}, \mathbf{e_i}) \\ &= \prod_{i=1}^{n} \underbrace{p(\mathbf{e_i}|\mathbf{x_i})}_{\text{extract}} \; \underbrace{p(y_i|\mathbf{e_i})}_{\text{classify}} \\ & \text{(assuming } \; y_i \perp \mathbf{x_i}|\mathbf{e_i}) \end{aligned}$$

This corresponds to the *extract, then classify* approach. Since both components of this likelihood function require extractions, supervised methods can only leverage m (out of n) training examples (Lehman et al., 2019). Unsupervised or semi-supervised extraction methods can still use all the document–level labels during training (Jain et al., 2020; Paranjape et al., 2020). Alternatively, we can factorize the likelihood as follows:

$$L = \prod_{i=1}^{n} \underbrace{p(y_i|\mathbf{x_i})}_{\text{classify}} \; \underbrace{p(\mathbf{e_i}|y_i, \mathbf{x_i})}_{\text{extract}}$$

The *classify, then extract* approach is amenable to weakly supervised learning, as we can optimize the classification objective for all n examples, and

Movie Review
I don't know what movie the critics saw, but it wasn't this one. The popular consensus among newspaper critics was that **this movie is unfunny and dreadfully boring** . In my personal opinion, they couldn't be more wrong. If you were expecting Airplane! - like laughs and Agatha Christie - intense mystery, then yes, this movie would be a disappointment. However, if you're just looking for **an enjoyable movie and a good time** , this is **one** to see ...
Lean, mean, escapist thrillers are a tough product to come by. **Most are unnecessarily complicated** , and others have no sense of expediency–the thrill-ride effect gets lost in the **cumbersome** plot. Perhaps the ultimate escapist thriller was the fugitive, which featured none of the flash-bang effects of today's market but rather a bread-and-butter, **textbook example of what a clever script and good direction is all about.** ...

Table 2: Non cherry-picked evidence extractions from our approach. We condition our extraction model on both the **positive** and the **negative** label. Our approach is able to tailor the extractions as per the conditioned label.

the extraction objective for m examples. We use BERT (Devlin et al., 2019) to model $p_\theta(y|\mathbf{x})$, and a linear-chain CRF (Lafferty et al., 2001) to model $p_\phi(\mathbf{e}|\mathbf{x}, y; \theta)$, where:

$$p_\phi(\mathbf{e}|y, \mathbf{x}) = \frac{1}{Z} \exp \left\{ \sum_{k=1}^{K} \phi_k f_k(e_t, e_{t-1}, x_t, y) \right\}$$

Here t indexes the input sequence, and Z is a normalization factor. Function $f(\cdot)$ extracts K features including both emission and transition features, and ϕ are the corresponding weights. The transition weights allow the CRF to model contiguity in the evidence tokens. We examine two types of emission features for a given token x_t in the input x, including (1) **BERT features** $(f_{\text{BERT}}(x)_t)$ where we encode the entire input sequence, and use the representation corresponding to token x_t;[3] and (2) **attention features** where we use the last layer attention values from different heads of the `[CLS]` token to the given token x_t. These features tie the classification and extraction architectures.

The *classify, then extract* approach also allows conditioning the evidence extraction model on the (predicted or oracle) **label** of the text document. For binary classification, one way to achieve this is to transform the existing emission features f to new features f' in the following manner:

$$f'_{2k}(e_t, e_{t-1}, x_t, y) = \begin{cases} f_k(e_t, e_{t-1}, x_t) & \text{if } y = 0 \\ 0 & \text{if } y = 1 \end{cases}$$

$$f'_{2k+1}(e_t, e_{t-1}, x_t, y) = \begin{cases} 0 & \text{if } y = 0 \\ f_k(e_t, e_{t-1}, x_t) & \text{if } y = 1 \end{cases}$$

This transformation allows us to use even indexed emission weights (ϕ_{2k}) for the first class, and odd indexed emission weights (ϕ_{2k+1}) for the second class. Similar transformations can be easily constructed for multi-class classification problems. During inference we use the predicted label $\hat{y}$ instead of the true label y. Using this formulation, emission features (and their corresponding weights) capture the association of each word with the extraction label (evidence or not) *and* the classification label. For instance, for binary sentiment analysis of movie reviews, the token "brilliant" is highly associated with the positive class, and if the review is (marked/predicted to be) positive, then the chances to select it as a part of the evidence increase. Inversely, if "brilliant" occurs in a negative review, the chances of selecting it decrease.

By conditioning the extraction models on the classification label, one can find supporting evidence tailored for each class (as one can see in Table 2). This can be especially useful when the input examples exhibit characteristics of multiple classes, or when classification models are less certain about their predictions. In such cases, examining the extractions for each class could help validate the model behaviour.

Implementation Details We train both the classification and extraction modules simultaneously. For evidence extraction, the emission features of the CRF include BERT representations or its attention values (depending upon the experiment). The same BERT model is also used for classification, thus the two tasks share the BERT parameters. We use the transformers library by Hugging Face (Wolf et al., 2019), and default optimization parameters for finetuning BERT.

4 Results and Discussion

Baselines We use several approaches that attempt to rationalize predictions as baselines for the evidence extraction task. These include: (i) the Pipeline approach (Lehman et al., 2019),

[3] Note that we share the BERT representations between the classification and extraction modules.

Approach	Sentiment Analysis		Propaganda Detection	
	Prediction (Accuracy)	Extraction (F1 score)	Prediction (F1 score)	Extraction (F1 score)
Pipeline approach$^\diamond$ (Lehman et al., 2019)	76.9	14.0	—	—
Information Bottleneck (IB)$^{\dagger\diamond}$ (Paranjape et al., 2020)	82.4	12.3	—	—
IB (semi-supervised, 25%)$^\diamond$ (Paranjape et al., 2020)	85.4	18.1	—	—
Top-k attention† (Jain et al., 2020)	93.1	27.7	65.8	27.4
Supervised attention (Zhong et al., 2019)	93.2	43.1	67.1	34.2
Our Methods				
Classify only (BERT)	93.1	—	65.8	—
Extract only (BERT-CRF)	—	42.6	—	39.1
Classify & Extract (BERT's Attention-CRF)	93.1	45.2	65.8	41.0
Classify & Extract (BERT-CRF)	93.3	45.4	64.1	**41.5**
Classify & Extract (BERT-CRF) w/ predicted labels	93.2	**46.3**	64.9	41.2
Classify & Extract (BERT-CRF) w/ oracle labels	93.3	46.8	64.9	45.0

Table 3: Evaluating different methods on two classification tasks that feature evidence annotations. The last row is an upper bound assuming access to the oracle label for conditioning. † denotes unsupervised approaches, and ◇ indicates sentence-level extraction methods, which can not be applied to the propaganda detection task as the input is only a single sentence. All the values are averaged across 5 seeds.

wherein the extraction and classification modules are pipelined, and individually trained with supervision; (ii) the Information Bottleneck approach (Paranjape et al., 2020) which extracts sentences from the input such that they have maximal mutual information (MI) with the output label, and minimal MI with the original input;[4] (iii) the FRESH approach (Jain et al., 2020), which extracts the top-k tokens with the highest attention scores (value of k is set to match the fraction of evidence tokens in the development set);[5] and (iv) Supervised attention, where attention is supervised to be uniformly high for tokens marked as evidence, and low otherwise (Zhong et al., 2019).

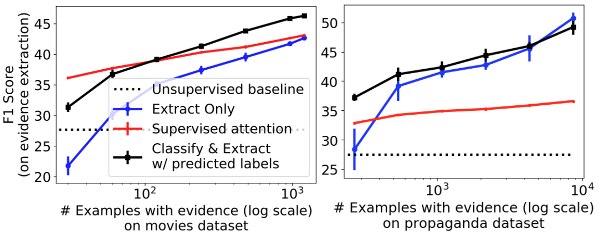
Setup We evaluate the different evidence extraction approaches on two text classification tasks: analyzing sentiment of movie reviews (Pang et al., 2002), and detecting propaganda techniques in news articles (Da San Martino et al., 2019). For the **sentiment analysis** task, we use the IMDb movie reviews dataset collected by Maas et al. (2011) comprising 25K movie reviews available for training, and 25K for development and testing. The dataset has disjoint sets of movies for training and testing. Additionally, we use 1.8K movie reviews with marked evidence spans collected by Zaidan et al. (2007). Of these 1.8K spans, we use 1.2K for training, and 300 each for development and testing. Note that here less than 5% of all the movie reviews are annotated for evidence, and the reviews are consistently long (with more than 600 words on an average), thus necessitating evidence to quickly verify the predictions.

For the task of **propaganda detection** in news articles, we use the binary sentence-level labels (propaganda or not), and token-level markings that support these labels. Similar to the sentiment dataset, we use token-level evidence markings for 5% of all the sentences. The total number of sentences in train, dev, and test sets are 10.8K, 1.7K, 4K respectively. Sentences without any propaganda content have no token-level markings.

Results We evaluate the predictions and their supporting evidences from different models. We compute the micro-averaged token-wise F1 score for the extraction task. From Table 3, we can clearly see that our approach outperforms other baseline methods on both the extraction tasks. The pipeline approach (Lehman et al., 2019) is unable to leverage a large pool of classification labels. Additionally, the pipeline and the Information Bottleneck approaches extract evidence at a sentence level, whereas the evidence markings are at a token level, which further explains their low scores. Further, the top-k attention baseline achieves a reasonable F1 score of 27.7 on the extraction task for sentiment analysis task, and 27.4 on the propaganda detection task, without any supervision. This result corroborates with findings of Jain et al.

[4] There exist trivial solutions to the Information Bottleneck objective when subset granularity is tokens instead of sentences. One such solution is when the extraction model extracts "." for the positive class, a "," for the negative class.

[5] Interestingly, Jain et al. (2020) find this simple thresholding approach to be better than other end-to-end approaches (Bastings et al., 2019; Lei et al., 2016)



(a) Sentiment Analysis (b) Propaganda detection

Figure 1: Mean and standard error of extraction scores with increasing amounts of evidence annotations.

(2020) who find attention scores to be good heuristics for extraction. Supervising attention with labeled extractions improves extraction F1 score on both tasks, which is inline with results in Zhong et al. (2019).

In our approach, the extraction model benefits from classification labels because of two factors: (i) sharing parameters between extraction and classification; and (ii) conditioning on the predicted $\hat{y}$ for extraction. These benefits are substantiated by comparing the extract only (BERT-CRF) approach with the classify & extract (BERT-CRF) method. The latter approach leads to improvements of 2.8 and 2.4 points for sentiment analysis and propaganda detection tasks, respectively. Conditioning on the predicted label improves the extractions by 0.9 points on the sentiment analysis task. For propaganda detection, we don't see an immediate benefit because many predicted labels are misclassified. However, upon using oracle labels, the extraction performance improves by 3.5 points.

When we lower the number of evidence annotations available during training, we discover (unsurprisingly) that the extraction performance degrades (Figure 1). For sentiment analysis, with less than 100 annotations, supervised attention performs the best, as no new parameters need to be trained. However, with over 100 training instances, classify & extract model outperforms this baseline, and is significantly better than the best unsupervised baseline. For propaganda detection, our approaches perform the best. As expected, the performance gap between extract only and classify & extract approach decreases with increase in available annotations.

5 Conclusion

We present a simple technique to supplement predictions with evidence by jointly modeling the text classification and evidence sequence labeling tasks. We show that conditioning the evidence extraction on the predicted label, in a *classify then extract* framework, leads to improved performance over baselines with as few as a hundred annotations. It also allows generating evidence for each label, which can enable stakeholders to better verify the correctness of predictions.

Acknowledgements

The authors are grateful to Anant Subramanian, Alankar Jain, Mansi Gupta, and Kundan Krishna for their insightful comments and suggestions. We also thank Highmark Health for their generous support of this research.

References

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news articles. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, EMNLP-IJCNLP 2019, Hong Kong, China.

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.

Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. 2003. The role of trust in automation reliance. *International journal of human-computer studies*, 58(6):697–718.

Jonathan L Herlocker, Joseph A Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. 2020. Learning to faithfully rationalize by construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *International Conference on Learning Representations (ICLR)*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *18th International Conference on Machine Learning 2001 (ICML 2001)*.

Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C. Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. 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 3705–3717, Minneapolis, Minnesota. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *Proceedings of the conference on Empirical methods in natural language processing*.

Zachary C Lipton. 2018. The mythos of model interpretability. *Queue*, 16(3):31–57.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Daniel S Weld and Gagan Bansal. 2019. The challenge of crafting intelligible intelligence. *Communications of the ACM*, 62(6):70–79.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the conference on Empirical methods in natural language processing*, pages 31–40.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267.

Ruiqi Zhong, Steven Shao, and Kathleen McKeown. 2019. Fine-grained sentiment analysis with faithful attention. *arXiv preprint arXiv:1908.06870*.