Zero-shot Sequence Labeling for Transformer-based Sentence Classifiers

Kamil Bujel
Department of Computing
Imperial College London
United Kingdom
kdb19@imperial.ac.uk

Helen Yannakoudakis
Department of Informatics
King's College London
United Kingdom
helen.yannakoudakis@kcl.ac.uk

Marek Rei
Department of Computing
Imperial College London
United Kingdom
marek.rei@imperial.ac.uk

Abstract

We investigate how sentence-level transformers can be modified into effective sequence labelers at the token level without any direct supervision. Existing approaches to zero-shot sequence labeling do not perform well when applied on transformer-based architectures. As transformers contain multiple layers of multi-head self-attention, information in the sentence gets distributed between many tokens, negatively affecting zero-shot token-level performance. We find that a soft attention module which explicitly encourages sharpness of attention weights can significantly outperform existing methods.

1 Introduction

Sequence labeling and sentence classification can represent facets of the same task at different granularities; for example, detecting grammar errors and predicting the grammaticality of sentences. Transformer-based architectures such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have been shown to achieve state-of-the-art results on both sequence labeling (Bell et al., 2019) and sentence classification (Sun et al., 2019) problems. However, such tasks are typically treated in isolation rather than within a unified approach.

In this paper, we investigate methods for inferring token-level predictions from transformer models trained only on sentence-level annotations. The ability to classify individual tokens without direct supervision opens possibilities for training sequence labeling models on tasks and datasets where only sentence-level or document-level annotation is available. In addition, attention-based architectures allow us to directly investigate what the model is learning and to quantitatively measure whether its *rationales* (supporting evidence) for particular input sentences match human expectations. While evaluating the *faithfulness* (Herman, 2017) of a model's rationale is still an open research question and up for debate (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; DeYoung et al., 2020; Jacovi and Goldberg, 2020; Atanasova et al., 2020), the methods explored here allow for measuring the *plausibility* (agreeability to human annotators; DeYoung et al. (2020)) of transformer-based models using existing sequence labeling datasets.

We evaluate and compare different methods for adapting pre-trained transformer models into zero-shot sequence labelers, trained using only gold sentence-level signal. Our experiments show that applying existing approaches (Rei and Søgaard, 2018) to transformer architectures is not straightforward – transformers already contain several layers of multi-head attention, distributing sentence-level information across many tokens, whereas the existing methods rely on all the information going through one central attention module. Approaches such as LIME (Ribeiro et al., 2016) for scoring word importance also struggle to infer correct token-level annotations in a zero-shot manner (e.g., it achieves only 2% F-score on one of our datasets). We find that a modified attention function is needed to allow transformers to better focus on individual important tokens and achieve a new state-of-the-art on zero-shot sequence labeling.

The contributions of this paper are fourfold:

- We present the first experiments utilizing (pre-trained) sentence-level transformers as zero-shot sequence labelers;
- We perform a systematic comparison of alternative methods for zero-shot sequence labeling on different datasets;
- We propose a novel modification of the attention function that significantly improves zero-shot sequence-labeling performance of transformers over the previous state of the art,

Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021), pages 195–205
Bangkok, Thailand (Online), August 6, 2021. ©2021 Association for Computational Linguistics

while achieving on-par or better results on sentence classification;

- We make our source code and models publicly available to facilitate further research in the field.[1]

2 Methods

We evaluate four different methods for turning sentence-level transformer models into zero-shot sequence labelers.

2.1 LIME

LIME (Ribeiro et al., 2016) generates local word-level importance scores through a meta-model that is trained on perturbed data generated by randomly masking out words in the input sentence. It was originally investigated in the context of Support Vector Machine (Hearst et al., 1998) text classifiers with unigram features.

We apply LIME to a RoBERTa model supervised as a sentence classifier and investigate whether its scores can be used for sequence labeling. We use RoBERTa's MASK token to mask out individual words and allow LIME to generate 5000 masked samples per sentence. The resulting explanation weights are then used as classification scores for each word, with the decision threshold fine-tuned based on the development set performance.

Thorne et al. (2019) found LIME to outperform attention-based approaches on the task of explaining NLI models. LIME was used to probe a LSTM-based sentence-pair classifier (Lan and Xu, 2018) by removing tokens from the premise and hypothesis sentences separately. The generated scores were used to perform binary classification of tokens, with the threshold based on F_1 performance on the development set. The token-level predictions were evaluated against human explanations of the entailment relation using the e-SNLI dataset (Camburu et al., 2018). LIME was found to outperform other methods, however, it was also 1000× slower than attention-based methods at generating these explanations.

2.2 Attention heads

The attention heads in a trained transformer model are designed to identify and combine useful information for a particular task. Clark et al. (2019) found that specific heads can specialize on different linguistic properties such as syntax and coreference. However, transformer models contain many layers with multiple attention heads, distributing the text representation and making it more difficult to identify token importance for the overall task.

Given a particular head, we can obtain an importance score for each token by averaging the attention scores from all the tokens that attend to it. In order to investigate the best possible setting, we report results for the attention head that achieves the highest token-level Mean Average Precision score on the development set.

2.3 Soft attention

Rei and Søgaard (2018) described a method for predicting token-level labels based on a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) architecture supervised at the sentence-level only. A dedicated attention module was integrated for building sentence representations, with its attention weights also acting as token-level importance scores. The architecture was found to outperform a gradient-based approach on the tasks of zero-shot sequence labeling for error detection, uncertainty detection, and sentiment analysis.

In order to obtain a single raw attention value $\widetilde{e}_i$ for each token, biLSTM output vectors were passed through a feedforward layer:

$$e_i = tanh(W_e h_i + b_e) \qquad \widetilde{e}_i = W_{\widetilde{e}} e_i + b_{\widetilde{e}} \qquad (1)$$

where e_i is the attention vector for token t_i; h_i is the biLSTM output for t_i; and $\widetilde{e}_i$ is the single raw attention value. W_e, b_e, $W_{\widetilde{e}}$, $b_{\widetilde{e}}$ are trainable parameters.

Instead of softmax or sparsemax (Martins and Astudillo, 2016), which would restrict the distribution of the scores, a soft attention based on sigmoid activation was used to obtain importance scores:

$$\widetilde{a}_i = \sigma(\widetilde{e}_i) \qquad a_i = \frac{\widetilde{a}_i}{\sum_{k=1}^{N} \widetilde{a}_k} \qquad (2)$$

where N is the number of tokens and σ is the logistic function. $\widetilde{a}_i$ shows the importance of a particular token and is in the range $0 \leq \widetilde{a}_i \leq 1$, independent of any other scores in the sentence; therefore, it can be directly used for sequence labeling with a natural threshold of 0.5. a_i contains the same information but is normalized to sum up to 1 over the whole sentence, making it suitable for attention weights when building the sentence representation.

[1] https://github.com/bujol12/bert-seq-interpretability

As a_i and $\widetilde{a}_i$ are directly tied, training the former through the sentence classification objective will also train the latter for the sequence labeling task.

The attention values were then used to obtain the sentence representation c by acting as weights for the biLSTM token outputs:

$$c = \sum_{i=0}^{N} a_i h_i \tag{3}$$

Finally, the sentence representation c was passed through the final feedforward layer, followed by a sigmoid to obtain the predicted score y for the sentence:

$$d = tanh(W_d c + b_d) \qquad y = \sigma(W_y d + b_y) \tag{4}$$

where d is the sentence vector, c is the sentence representation, and y is the sentence prediction score. W_d, b_d, W_y, b_y are all trainable parameters.

We adapt this approach to the transformer models by attaching a separate soft attention module on top of the token-level output representations. This effectively ignores the CLS token, which is commonly used for sentence classification, and instead builds a new sentence representation from the token representations, which replace the previously used biLSTM outputs:

$$e_i = tanh(W_e T_i + b_e) \quad c = \sum_{i=0}^{N} a_i T_i \tag{5}$$

where T_i is the contextualized embedding for token t_i. A diagram of the model architecture is included in Appendix F.

Commonly used tokenizers for transformer models split words into subwords, while sequence labeling datasets are annotated at the word level. We find that taking the maximum attention value over all the subwords as the word-level importance score produces good results on the development sets. For a word w_i split into tokens $[t_j, ..., t_m]$, where $j, m \in [1, N]$, the resulting final word importance score r_i is then given by:

$$r_i = max(\{\widetilde{a}_j, \widetilde{a}_{j+1}, ..., \widetilde{a}_m\}) \tag{6}$$

During training, we optimize sentence-level binary cross-entropy as the main objective function:

$$L_1 = \frac{\sum_j CrossEntropy(y^{(j)}, \widetilde{y}^{(j)})}{|y|} \tag{7}$$

where $y^{(j)}$ and $\widetilde{y}^{(j)}$ are the predicted sentence classification logits and the gold label for the j^{th} sentence respectively. We also adopt the additional loss functions from Rei and Søgaard (2018), which encourage the attention weights to behave more like token-level classifiers:

$$L_2 = \frac{\sum_j (min_j(\widetilde{a}_i) - 0)^2}{|y|} \tag{8}$$

$$L_3 = \frac{\sum_j (max_j(\widetilde{a}_i) - \widetilde{y}^{(j)})^2}{|y|} \tag{9}$$

Eq. 8 optimizes the minimum unnormalized attention to be 0 and therefore incentivizes the model to only focus on some, but not all words; Eq. 9 ensures that some attention weights are close to 1 if the overall sentence is classified as positive. We then jointly optimize these three loss functions using a hyperparameter γ: $L = L_1 + \gamma(L_2 + L_3)$.

2.4 Weighted soft attention

Our experiments show that, when combined with transformer-based models, the soft attention method tends to spread out the attention too widely. Instead of focusing on specific important words, the model broadly attends to the whole sentence. Figures 3 and 4 in Appendix A present examples demonstrating such behaviour. As transformers contain several layers of attention, with multiple heads in each layer, the information in the sentence gets distributed across all tokens before it reaches the soft attention module at the top.

To improve this behaviour and incentivize the model to direct information through a smaller and more focused set of tokens, we experiment with a weighted soft attention:

$$a_i = \frac{\tilde{a}_i^{\beta}}{\sum_{k=1}^{N} \tilde{a}_k^{\beta}} \tag{10}$$

where β is a hyperparamete and where values $\beta > 1$ make the weight distribution sharper, allowing the model to focus on a smaller number of tokens. We experiment with values of $\beta \in \{1, 2, 3, 4\}$ on the development sets and find $\beta = 2$ to significantly improve token labeling performance without negatively affecting sentence classification results.

3 Datasets

We investigate the performance of these methods as zero-shot sequence labelers using three different datasets. Gold token-level annotation in these

	FCE			BEA 2019			CoNLL 2010		
	Sent F_1	F_1	MAP	Sent F_1	F_1	MAP	Sent F_1	F_1	MAP
Random baseline	-	23.19	33.95	-	16.73	27.01	-	1.63	14.15
RoBERTa	84.51	-	-	83.66	-	-	86.66	-	-
Rei and Søgaard (2018)	84.75	28.73	48.56	81.27	18.53	31.69	84.16	**72.42**	87.82
LIME	84.51	24.60	37.90	83.66	2.09	31.41	86.66	57.14	78.44
Attention heads	84.51	24.34	48.04	83.66	19.69	40.55	86.66	25.64	79.82
Soft attention	**85.62**	32.16	48.90	83.41	22.92	35.79	86.25	8.45	20.04
Weighted soft attention	**85.62**	**33.31**	**53.91**	**83.68**	**24.35**	**41.07**	**87.20**	67.28	**91.18**

Table 1: Results on FCE, BEA 2019 and CoNLL 2010. *Sent* F_1 refers to F-measure on the sentence classification task; F_1 refers to token-level classification performance; *MAP* is the token-level Mean Average Precision.

datasets is used for evaluation; however, the models are trained using sentence-level labels only.

The **CoNLL 2010** shared task (Farkas et al., 2010)[2] focuses on the detection of uncertainty cues in natural language text. The dataset contains $19,542$ examples with both sentence-level uncertainty labels and annotated keywords indicating uncertainty. We use the train/test data from the task and randomly choose 10% of the training set for development.

We also evaluate on the task of grammatical error detection (GED) – identifying which sentences are grammatically incorrect (i.e., contain at least one grammatical error). The First Certificate in English dataset **FCE** (Yannakoudakis et al., 2011) consists of essays written by non-native learners of English, annotated for grammatical errors. We use the train/dev/test splits released by Rei and Yannakoudakis (2016) for sequence labeling, with a total of $33,673$ sentences.

In addition, we evaluate on the Write & Improve (Yannakoudakis et al., 2018) and LOCNESS (Granger, 1998) GED dataset[3] ($38,692$ sentences) released as part of the **BEA 2019** shared task (Bryant et al., 2019). It contains English essays written in response to varied topics and by English learners from different proficiency levels, as well as native English speakers. As the gold test set labels are not publicly available, we evaluate on the released development set and use 10% of the training data for tuning[4]. For both GED datasets, we train the model to detect grammatically incorrect sentences and evaluate how well the methods can identify individual tokens that have been annotated as errors.

4 Experimental setup

We use the pre-trained RoBERTa-base (Liu et al., 2019) model, made available by HuggingFace (Wolf et al., 2020), as our transformer architecture. Following Mosbach et al. (2021), transformer models are fine-tuned for 20 epochs, and the best performing checkpoint is then chosen based on sentence-level performance on the development set. Each experiment is repeated with 5 different random seeds and the averaged results are reported. The average duration of training on Nvidia GeForce RTX 2080Ti was 1 hour. Significance testing is performed with a two-tailed paired t-test and $a = 0.05$. Hyperparameteres are tuned on the development set and presented in Appendices B and C.

The LIME and attention head methods provide only a score without a natural decision boundary for classification. Therefore, we choose their thresholds based on the token-level F_1-score on the development set. In contrast, the soft attention and weighted soft attention methods do not require such additional tuning that uses token-level labels.

5 Results

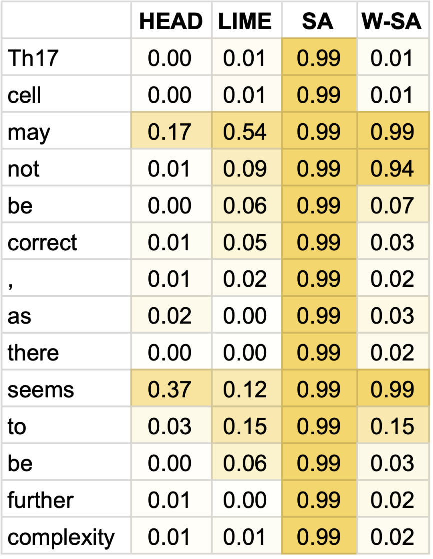
The results are presented in Table 1. Each model is trained as a sentence classifier and then evaluated as a token labeler. The challenge of the zero-shot sequence-labeling setting lies in the fact that the models are trained without utilizing any gold token-level signal; nevertheless, some methods perform considerably better than others. For reference, we also include a random baseline, which samples token-level scores from the standard uniform distribution; a RoBERTa model supervised as a sentence classifier only; and the model from Rei and Søgaard (2018) based on BiLSTMs.

We report the F_1-measure on the token level along with Mean Average Precision (MAP) for returning positive tokens. The MAP metric views the task as a ranking problem and therefore removes

[2] https://rgai.sed.hu/node/118

[3] https://www.cl.cam.ac.uk/research/nl/bea2019st/

[4] https://github.com/bujol12/bert-seq-interpretability/blob/master/dev_indices_train_ABC.txt



	HEAD	LIME	SA	W-SA
Th17	0.00	0.01	0.99	0.01
cell	0.00	0.01	0.99	0.01
may	0.17	0.54	0.99	0.99
not	0.01	0.09	0.99	0.94
be	0.00	0.06	0.99	0.07
correct	0.01	0.05	0.99	0.03
,	0.01	0.02	0.99	0.02
as	0.02	0.00	0.99	0.03
there	0.00	0.00	0.99	0.02
seems	0.37	0.12	0.99	0.99
to	0.03	0.15	0.99	0.15
be	0.00	0.06	0.99	0.03
further	0.01	0.00	0.99	0.02
complexity	0.01	0.01	0.99	0.02

Figure 1: Example word-level importance scores r_i (Eq. 6) of different methods applied to an excerpt from the CoNLL10 dataset. *HEAD* corresponds to attention heads; *SA* to soft attention; and *W-SA* to weighted soft attention. We can observe how *W-SA* is the only method that correctly assigns substantially higher weights to the 'may' and 'seems' uncertainty cues.

the dependence on specific classification thresholds. In addition, we report the F_1-measure on the main sentence-level task to ensure the proposed methods do not have adverse effects on sentence classification performance. Precision and recall values are included in Appendix E.

LIME has relatively low performance on FCE and BEA 2019, while it achieves somewhat higher results on CoNLL 2010. Comparing the MAP scores, the attention head method performs substantially better, especially considering that it is much more lightweight and requires no additional computation. Nevertheless, both of these methods rely on using some annotated examples to tune their classification threshold, which precludes their application in a truly zero-shot setting.

Combining the soft attention mechanism with the transformer architecture provides some improvements over the previous methods, while also improving over Rei and Søgaard (2018). A notable exception is the CoNLL 2010 dataset where this method achieves only 8% F_1 and 20% MAP. Error analysis revealed that this is due to the transformer representations spreading attention scores evenly between a large number of tokens, as observed in Figure 1. Uncertainty cues in CoNLL 2010 can span across whole sentences (e.g., *'Either ... or ...'*), with such examples encouraging the model to distribute information even further.

The weighted soft attention modification addresses this issue and considerably improves performance across all metrics on all datasets. Compared to the non-weighted version of the soft attention method, applying the extra weights leads to a significant improvement in terms of MAP, with a minimum of 5.01% absolute gain on FCE. The improvements are also statistically significant compared to the current state of the art (Rei and Søgaard, 2018): 5.35% absolute improvement on FCE; 9.38% on BEA 2019; and 3.36% on CoNLL 2010. While the F_1 on CoNLL 2010 is slightly lower, the MAP score is higher, indicating that the model has difficulty finding an optimal decision boundary, but nevertheless provides a better ranking. In future work, the weighted soft attention method for transformers could potentially be combined with token supervision in order to train robust multi-level models (Barrett et al., 2018; Rei and Søgaard, 2019).

6 Conclusion

We investigated methods for inferring token-level predictions from transformer models trained only on sentence-level annotations. Experiments showed that previous approaches designed for LSTM architectures do not perform as well when applied to transformers. As transformer models already contain multiple layers of multi-head attention, the input representations get distributed between many tokens, making it more difficult to identify the importance of each individual token. LIME was not able to accurately identify target tokens, while the soft attention method primarily assigned equal attention scores across most words in a sentence. Directly using the scores from the existing attention heads performed better than expected, but required some annotated data for tuning the decision threshold. Modifying the soft attention module with an explicit sharpness constraint on the weights was found to encourage more distinct predictions, significantly improving token-level results.

Acknowledgments

We would like to thank James Thorne for his assistance in setting up the LIME experiments. Kamil Bujel was funded by the Undergraduate Research Opportunities Programme Bursary from the Department of Computing at Imperial College London.

References

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. 2018. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 302–312, Brussels, Belgium. Association for Computational Linguistics.

Samuel Bell, Helen Yannakoudakis, and Marek Rei. 2019. Context is key: Grammatical error detection with contextual word representations. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 103–115, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *NeurIPS*, pages 9560–9572.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. 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.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The CoNLL-2010 shared task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning – Shared Task*, pages 1–12, Uppsala, Sweden. Association for Computational Linguistics.

Sylviane Granger. 1998. *The computer learner corpus: A versatile new source of data for SLA research*. Longman.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Bernease Herman. 2017. The promise and peril of human evaluation for model interpretability. *arXiv preprint arXiv:1711.07414*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Wuwei Lan and Wei Xu. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3890–3902, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623. PMLR.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Marek Rei and Anders Søgaard. 2018. Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. 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 293–302, New Orleans, Louisiana. Association for Computational Linguistics.

Marek Rei and Anders Søgaard. 2019. Jointly learning to label sentences and tokens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6916–6923.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1191, Berlin, Germany. Association for Computational Linguistics.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2019. Generating token-level explanations for natural language inference. 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 963–969, Minneapolis, Minnesota. Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

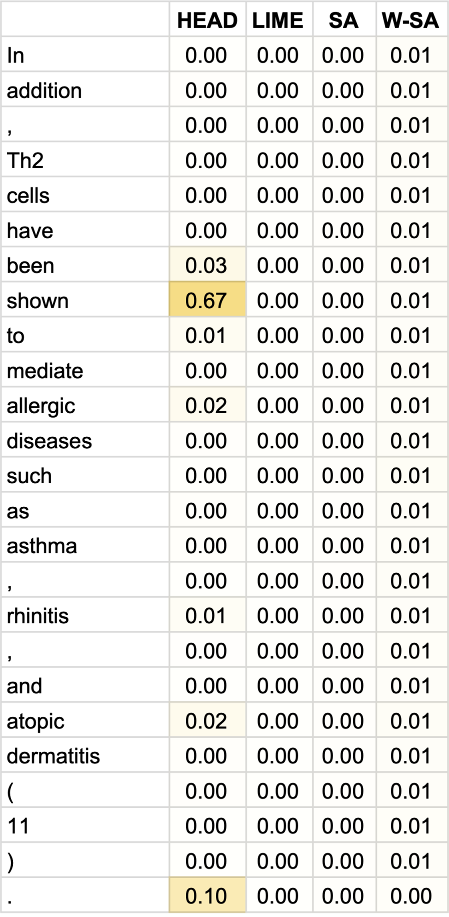
Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest,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

Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an automated writing placement system for esl learners. *Applied Measurement in Education*, 31(3):251–267.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

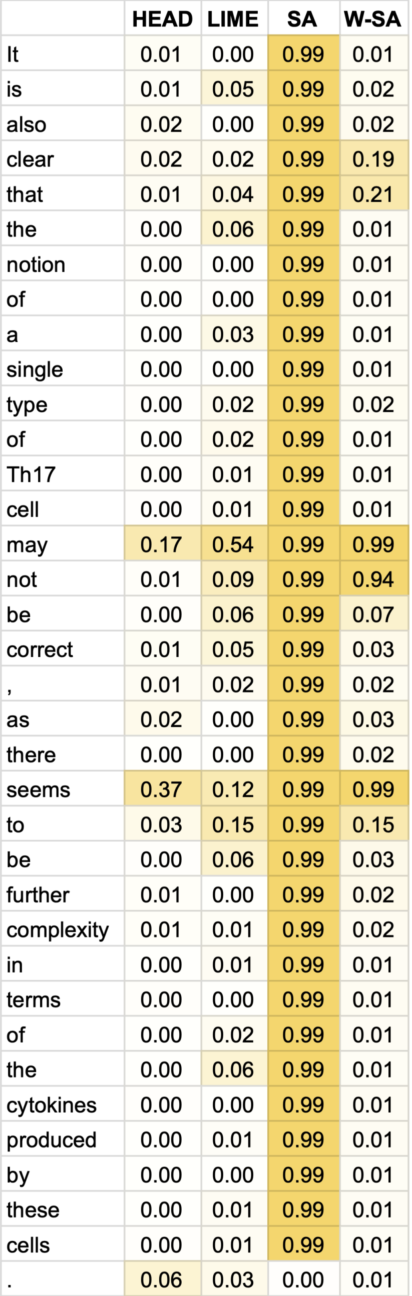
A Example word-level predictions

We present samples of word-level predictions (word-level importance scores r_i, Eq. 6) to illustrate differences between methods. In the figures that follow, *HEAD* refers to attention heads, SA to soft attention, and *W-SA* to weighted soft attention.



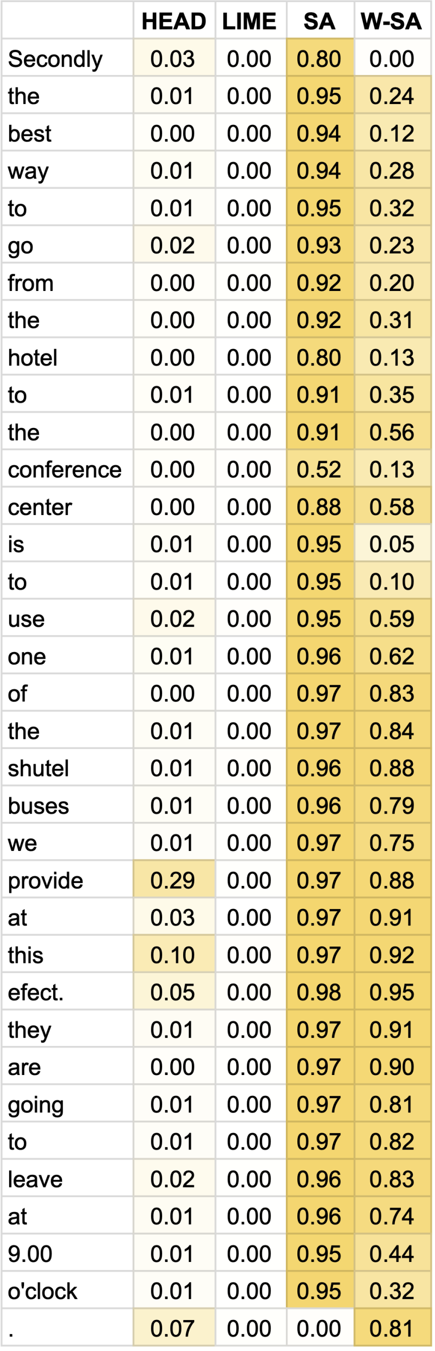
	HEAD	LIME	SA	W-SA
In	0.00	0.00	0.00	0.01
addition	0.00	0.00	0.00	0.01
,	0.00	0.00	0.00	0.01
Th2	0.00	0.00	0.00	0.01
cells	0.00	0.00	0.00	0.01
have	0.00	0.00	0.00	0.01
been	0.03	0.00	0.00	0.01
shown	0.67	0.00	0.00	0.01
to	0.01	0.00	0.00	0.01
mediate	0.00	0.00	0.00	0.01
allergic	0.02	0.00	0.00	0.01
diseases	0.00	0.00	0.00	0.01
such	0.00	0.00	0.00	0.01
as	0.00	0.00	0.00	0.01
asthma	0.00	0.00	0.00	0.01
,	0.00	0.00	0.00	0.01
rhinitis	0.01	0.00	0.00	0.01
,	0.00	0.00	0.00	0.01
and	0.00	0.00	0.00	0.01
atopic	0.02	0.00	0.00	0.01
dermatitis	0.00	0.00	0.00	0.01
(	0.00	0.00	0.00	0.01
11	0.00	0.00	0.00	0.01
)	0.00	0.00	0.00	0.01
.	0.10	0.00	0.00	0.00

Figure 2: CoNLL 2010 negative sentence (without uncertainty cues). We can clearly see that most methods correctly put weights close to 0 for all words, except from HEAD, which focuses on 'shown' and '.'. We surmise this is due to the fact that, for HEAD, weights over the whole sentence have to sum up to 1.



	HEAD	LIME	SA	W-SA
It	0.01	0.00	0.99	0.01
is	0.01	0.05	0.99	0.02
also	0.02	0.00	0.99	0.02
clear	0.02	0.02	0.99	0.19
that	0.01	0.04	0.99	0.21
the	0.00	0.06	0.99	0.01
notion	0.00	0.00	0.99	0.01
of	0.00	0.00	0.99	0.01
a	0.00	0.03	0.99	0.01
single	0.00	0.00	0.99	0.01
type	0.00	0.02	0.99	0.02
of	0.00	0.02	0.99	0.01
Th17	0.00	0.01	0.99	0.01
cell	0.00	0.01	0.99	0.01
may	0.17	0.54	0.99	0.99
not	0.01	0.09	0.99	0.94
be	0.00	0.06	0.99	0.07
correct	0.01	0.05	0.99	0.03
,	0.01	0.02	0.99	0.02
as	0.02	0.00	0.99	0.03
there	0.00	0.00	0.99	0.02
seems	0.37	0.12	0.99	0.99
to	0.03	0.15	0.99	0.15
be	0.00	0.06	0.99	0.03
further	0.01	0.00	0.99	0.02
complexity	0.01	0.01	0.99	0.02
in	0.00	0.01	0.99	0.01
terms	0.00	0.00	0.99	0.01
of	0.00	0.02	0.99	0.01
the	0.00	0.06	0.99	0.01
cytokines	0.00	0.00	0.99	0.01
produced	0.00	0.01	0.99	0.01
by	0.00	0.00	0.99	0.01
these	0.00	0.01	0.99	0.01
cells	0.00	0.01	0.99	0.01
.	0.06	0.03	0.00	0.01

Figure 3: CoNLL 2010 positive sentence (with uncertainty cues). We can observe that HEAD correctly identifies both of the uncertainty cues: 'may' and 'seems'; however the weight for 'may' is quite low. Similarly, LIME identifies both tokens, but the weight for 'seems' is particularly low (lower than for 'to'). SA simply assigns high weights to all words. W-SA focuses primarily on the two uncertainty cue words; however, it also incorrectly focuses on 'not'.



	HEAD	LIME	SA	W-SA
Secondly	0.03	0.00	0.80	0.00
the	0.01	0.00	0.95	0.24
best	0.00	0.00	0.94	0.12
way	0.01	0.00	0.94	0.28
to	0.01	0.00	0.95	0.32
go	0.02	0.00	0.93	0.23
from	0.00	0.00	0.92	0.20
the	0.00	0.00	0.92	0.31
hotel	0.00	0.00	0.80	0.13
to	0.01	0.00	0.91	0.35
the	0.00	0.00	0.91	0.56
conference	0.00	0.00	0.52	0.13
center	0.00	0.00	0.88	0.58
is	0.01	0.00	0.95	0.05
to	0.01	0.00	0.95	0.10
use	0.02	0.00	0.95	0.59
one	0.01	0.00	0.96	0.62
of	0.00	0.00	0.97	0.83
the	0.01	0.00	0.97	0.84
shutel	0.01	0.00	0.96	0.88
buses	0.01	0.00	0.96	0.79
we	0.01	0.00	0.97	0.75
provide	0.29	0.00	0.97	0.88
at	0.03	0.00	0.97	0.91
this	0.10	0.00	0.97	0.92
efect.	0.05	0.00	0.98	0.95
they	0.01	0.00	0.97	0.91
are	0.00	0.00	0.97	0.90
going	0.01	0.00	0.97	0.81
to	0.01	0.00	0.97	0.82
leave	0.02	0.00	0.96	0.83
at	0.01	0.00	0.96	0.74
9.00	0.01	0.00	0.95	0.44
o'clock	0.01	0.00	0.95	0.32
.	0.07	0.00	0.00	0.81

Figure 4: FCE positive sentence (contains grammatical errors). We can see that both LIME and HEAD struggle to assign informative and/or useful weights to the words. All SA weights are relatively high, with small variations in value. We can see that squaring (W-SA) leads to more well-defined weights over the whole sentence, with high weights mainly observed in the second part of the sentence, which is the one that contains incorrect words. However, on this dataset, even W-SA struggles to correctly identify which words precisely are incorrect.

B Hyperparameters

Name	Value
γ	0.1
max seq length	128
per device train batch size	16
per device eval batch size	64
warmup ratio	0.1
learning rate	2e-5
weight decay	0.1
adam epsilon	1e-7
hidden layer dropout	0.1
soft attention layer size	100
soft attention hidden size	300
initializer	glorot

Table 2: Model hyperparameters.

C Word-level prediction thresholds

Dataset	Method	Threshold
CoNLL 2010	LIME	0.200
	Random baseline	0.500
	Attention heads	0.320
	Rei and Søgaard (2018)	0.500
	Soft attention	0.500
	Weighted soft attention	0.500
FCE	LIME	0.001
	Random baseline	0.500
	Attention heads	0.080
	Rei and Søgaard (2018)	0.500
	Soft attention	0.500
	Weighted soft attention	0.500
BEA 2019	LIME	0.010
	Random baseline	0.500
	Attention heads	0.080
	Rei and Søgaard (2018)	0.500
	Soft attention	0.500
	Weighted soft attention	0.500

Table 3: Word-level thresholds above which a word is classified as positive.

D Validation set results

Dataset	Method	Sent F_1
CoNLL 2010	LIME	91.77
	RoBERTa	91.77
	Attention heads	91.77
	Soft attention	92.12
	Weighted soft attention	91.82
FCE	LIME	84.49
	RoBERTa	84.49
	Attention heads	84.49
	Soft attention	84.82
	Weighted soft attention	85.56
BEA 2019	LIME	83.65
	RoBERTa	83.65
	Attention heads	83.65
	Soft attention	83.47
	Weighted soft attention	83.64

Table 4: Mean sentence-level F_1 score on the development set, averaged over 5 runs.

E Full test set results

	FCE		
	Sent F_1	Sent P	Sent R
Random baseline	-	-	-
RoBERTa	84.51	84.25	**84.93**
Rei and Søgaard (2018)	84.75	-	-
LIME	84.51	84.25	**84.93**
Attention heads	84.51	84.25	**84.93**
Soft attention	**85.62**	**86.92**	84.42
Weighted soft attention	**85.62**	86.88	84.45

Table 5: Sentence-level results: *P*, *R* and F_1 refer to Precision, Recall and F-measure respectively on the positive class.

	BEA 2019		
	Sent F_1	Sent P	Sent R
Random baseline	-	-	-
RoBERTa	83.66	**82.29**	85.15
Rei and Søgaard (2018)	81.27	-	-
LIME	83.66	**82.29**	85.15
Attention heads	83.66	**82.29**	85.15
Soft attention	83.41	81.47	85.54
Weighted soft attention	**83.68**	79.95	**87.91**

Table 6: Sentence-level results: *P*, *R* and F_1 refer to Precision, Recall and F-measure respectively on the positive class.

	CoNLL 2010		
	Sent F_1	Sent P	Sent R
Random baseline	-	-	-
RoBERTa	86.66	84.90	**88.63**
Rei and Søgaard (2018)	84.16	-	-
LIME	86.66	84.90	**88.63**
Attention heads	86.66	84.90	**88.63**
Soft attention	86.25	85.75	86.89
Weighted soft attention	**87.20**	**89.17**	85.37

Table 7: Sentence-level results: *P*, *R* and F_1 refer to Precision, Recall and F-measure respectively on the positive class.

	FCE			
	P	R	F_1	MAP
Random baseline	15.11	49.81	23.19	33.95
RoBERTa	-	-	-	-
Rei and Søgaard (2018)	**29.16**	29.04	28.73	48.56
LIME	19.06	34.70	24.60	37.90
Attention heads	26.67	22.38	24.34	48.04
Soft attention	19.84	**85.38**	32.16	48.90
Weighted soft attention	20.76	85.36	**33.31**	**53.91**

Table 8: Token-level results: *P*, *R* and F_1 refer to Precision, Recall and F-measure respectively on the positive class. *MAP* is the Mean Average Precision at the token-level.

	BEA 2019			
	P	R	F_1	MAP
Random baseline	10.05	50.00	16.73	27.01
RoBERTa	-	-	-	-
Rei and Søgaard (2018)	10.93	61.63	18.53	31.69
LIME	13.49	1.13	2.09	31.41
Attention heads	**18.48**	21.07	19.69	40.55
Soft attention	13.20	**87.19**	22.92	35.79
Weighted soft attention	14.20	85.49	**24.35**	**41.07**

Table 9: Token-level results: *P*, *R* and F_1 refer to Precision, Recall and F-measure respectively on the positive class. *MAP* is the Mean Average Precision at the token-level.

	CoNLL 2010			
	P	R	F_1	MAP
Random baseline	0.83	49.70	1.63	14.15
RoBERTa	-	-	-	-
Rei and Søgaard (2018)	**78.99**	67.06	**72.42**	87.82
LIME	63.25	52.11	57.14	78.44
Attention heads	22.33	30.11	25.64	79.82
Soft attention	4.48	**86.14**	8.45	20.04
Weighted soft attention	58.80	78.89	67.28	**91.18**

Table 10: Token-level results: *P*, *R* and F_1 refer to Precision, Recall and F-measure respectively on the positive class. *MAP* is the Mean Average Precision at the token-level.

F Weighted soft attention architecture

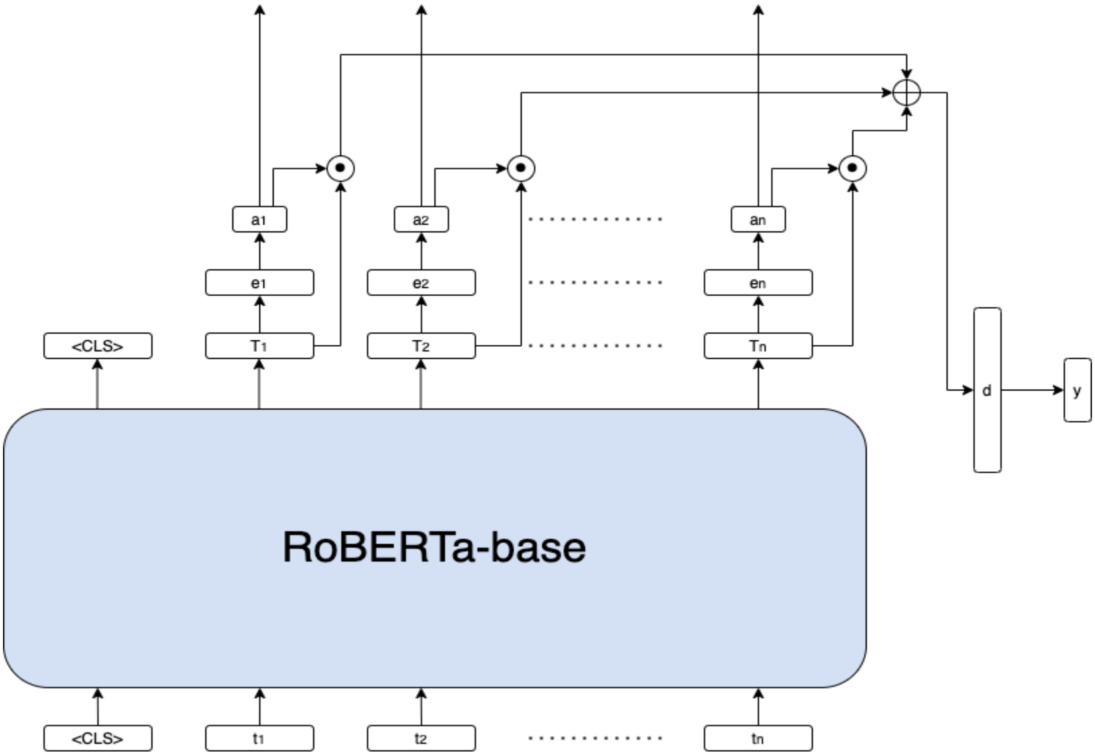


Figure 5: Architecture of our proposed weighted soft attention model. $[t_1, t_2, ..., t_n]$ represent the tokenized input sentence, while $[T_1, T_2, ..., T_n]$ are the resulting contextual embeddings. $[e_1, e_2, ..., e_n]$ are attention vectors, and $[a_1, a_2, ..., a_n]$ are normalized attention weights. d represents the output vector and y the final output logits.