Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models

Reda Yacouby
Amazon Alexa
redaya@amazon.com

Dustin Axman
Amazon Alexa
dax@amazon.com

Abstract

In pursuit of the perfect supervised NLP classifier, razor thin margins and low-resource test-sets can make modeling decisions difficult. Popular metrics such as Accuracy, Precision, and Recall are often insufficient as they fail to give a complete picture of the model's behavior. We present a probabilistic extension of Precision, Recall, and F1 score, which we refer to as confidence-Precision (cPrecision), confidence-Recall (cRecall), and confidence-F1 (cF1) respectively. The proposed metrics address some of the challenges faced when evaluating large-scale NLP systems, specifically when the model's confidence score assignments have an impact on the system's behavior. We describe four key benefits of our proposed metrics as compared to their threshold-based counterparts. Two of these benefits, which we refer to as *robustness to missing values* and *sensitivity to model confidence score assignments* are self-evident from the metrics' definitions; the remaining benefits, *generalization*, and *functional consistency* are demonstrated empirically.

1 Introduction

Supervised machine learning classifiers are typically trained to minimize error. This error is evaluated using one or multiple metrics, the choice of which has been a continuous debate in research and industry for multiple decades (Dinga et al., 2019; Brier, 1950). Many criteria need to be considered when choosing a metric, including but not limited to: interpretability, computational cost, differentiability, and popularity in a specific field. As an example, a typical workflow of model development is to use a loss function such as cross-entropy or hinge loss during training for weight optimization, then use an easily interpretable metric such as Accuracy, Precision, or Recall when testing the model against a holdout sample of examples. This is because the mentioned loss functions are differentiable convex functions, enabling optimization algorithms such as gradient descent to find minima with reasonable computational cost. In contrast, the test-set evaluation metrics are often required to be easy to relate to the real-world problem the classifier is designed to help solve, in order to give a concrete idea of performance or success to the stakeholders.

Essentially, all the criteria mentioned serve the same underlying purpose of driving modeling decisions. The heterogeneous nature of model evaluation illustrates how there could be no universal criteria for driving model decisions, or so-called "best metric", as each criterion could be advantageous under specific operating conditions (Hernández-Orallo et al., 2012), or even preferred by stakeholders for reasons that do not need to be scientifically driven (such as interpretability and business purposes).

In the Natural Language Processing (NLP) industry, new challenges have risen in the past few years in terms of performance evaluation, due to the complexity and scalable design of modern NLP systems such as those powering Google Assistant, Amazon Alexa, or Apple's Siri (Sarikaya, 2017). Such systems are built to support devices with a potentially limitless number of functionalities, as reflected by the Alexa Skill Developer Toolkit and Google Actions, allowing external developers to add additional functionality to the NLP system, supporting new phrases and therefore increasing the number of choices the system needs to disambiguate between.

This sharp rise in scale and complexity has made the most commonly used metrics (Accuracy, Precision, Recall, F1 score) insufficient in depicting a comprehensive picture of the impact introduced by changes in these systems. A key reason behind this gap is that classification models typically

Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP), pages 79–91,
November 20, 2020. ©2020 Association for Computational Linguistics

output an n-best list of model predictions, each associated with a confidence score (or probability score), and while simple systems (and most academic use-cases) only consider the highest-score prediction, more elaborate systems tend to leverage further information from the n-best to drive decisions. Metrics such as Accuracy, Precision and Recall simply compare the highest-score prediction with the test reference, ignoring the rest of the n-best output, while this ignored information often does impact the behavior of the NLP system.

We provide 2 cases that exemplify the case of an NLP system being impacted by changes in the n-best output which are ignored by popular metrics:

1. **Arbitration:** Some specific criteria could be used to arbitrate between the n-best predictions rather than always choosing the highest-score prediction. For example, if the top prediction is un-actionable by the system (e.g. results in an error) and the second-best prediction meets some defined criteria, the system could fall-back to that prediction. In the case of a vocal assistant an example would be asking a TV to "play frozen", and the NLP model recognizes it as a request to play song called "frozen" as its top prediction, while the homonymous movie is the second-best prediction. The system could arbitrate and decide to use the second prediction specifically because the request was spoken to a TV, rather than a music player.

2. **Error correction:** It is common for large-scale NLP systems to be multi-step, having domain-specialized models receive the output of an upstream NLP model as input, then attempting to correct potential mistakes. As an example, Named Entity Recognition (NER) in the Shopping domain is challenging for general purpose NLP models due to the large size of product catalogs and potentially ambiguous product names. A downstream Shopping-specific NLP model can be applied on the upstream model's n-best for error correction, potentially re-ranking the n-best and adjusting confidence scores.

Many evaluation metrics capable of measuring changes in confidence score assignments already exist. In this document we will use the taxonomy introduced by Ferri et al. (2009), classifying metrics into 3 categories:

- Threshold-based metrics, using a qualitative understanding of error, such as Accuracy, Precision, and Recall.

- Rank-based metrics, which evaluate how well the models ranks the examples. The Area Under the ROC Curve (AUC) is the most widely used in this category.

- Probabilistic metrics, using a probabilistic understanding of error, as they consider the confidence scores assigned by the models in their measurements. Among these are Brier-score and Cross-entropy.

Among these categories, probabilistic metrics have the potential to fill the evaluation gaps we described. In this paper we are proposing a probabilistic extension of threshold-based metrics. The goal is to introduce advantages of probabilistic metrics while retaining the relatability of threshold-based metrics to the real-world operating cost function of the models, allowing for decision making that is both scientifically reliable and tied to the stakeholder's interests. We describe in Section 3 why other probabilistic metrics are not sufficient to fill the evaluation gaps we are addressing with the newly proposed metrics.

One of the primary benefits of probabilistic metrics is their ability to function more consistently in test-data sparse scenarios. It was demonstrated empirically by Wang et al. (2013) and Dinga et al. (2019) that probabilistic metrics are more reliable in discriminating between models, since they leverage the most information from the model's output. This does not necessarily make them *better* metrics, as we stated earlier how modeling decisions are closely tied to operating conditions, but allows them to be more data-efficient (require less data to reach statistically significant results). Recent developments in Transfer Learning (Pan and Yang, 2010; Conneau et al., 2020) demonstrated impressive ability to learn from small training sets (often referred to as Few-Shot Learning), showing a wide NLP community interest in improving data-efficiency during model training, but we have not found any publication related to data-efficient model testing. Usually the lack of training data would also imply a lack of test data, as they would be caused by the same underlying factor (expensive data collection and/or labelling, low-resource language), which highlights the value in developing ways to compare models with minimal test data requirements. As part of our investigation in this subject, we empirically show that our proposed metrics are more data efficient than their threshold-based counterparts, as they allow for modeling decisions with smaller test-sets.

On a side note, some of the cases in which model confidence assignments are used in production require the scores to be probabilistically calibrated, as they are interpreted by the users as probabilities of events happening (e.g. disease prevention, weather forecasts). Probabilistic calibration refers to the reliability of the scores in reflecting the true probability of the predictions being correct (e.g. if a calibrated model predicted in *n* cases that event *X* will happen with probability *p*, then event X should happen in approximately *p***n* of those cases). The proposed metrics do not evaluate for probabilistic calibration. For such use-cases we suggest the combined usage of a probability calibration measure (e.g. the Expected Calibration Error (Guo et al., 2017), the reliability component of Brier-score (Murphy, 1973)) along with the proposed metrics, for a thorough evaluation of both performance and calibration.

In this document, we describe four benefits of the proposed metrics. In comparison with their threshold-based counterparts, our metrics:

A. Have an equal or lower likelihood of being NaN (*Robustness to NaN values*).

B. Are sensitive to changes in the model's confidence scores across the model's full n-best output (*Sensitivity to model confidence score assignments*).

C. Have lower variance, making their point estimates more generalizable to unseen data, and allowing for better discriminancy between models (*Generalization* hypothesis)

D. Provide the same ranking of performance of candidate models as their threshold-based counterpart's population value in the majority of cases (*Functional Consistency* hypothesis).

The first two are easily deduced from the metrics' definitions. The third and fourth benefits are demonstrated empirically in Section 6.

2 Definitions

cPrecision, cRecall, and cF1-Score have the same mathematical formulations as Precision, Recall, and F1-Score, respectively, with the only difference being the usage of continuous (as opposed to binary) definitions of Positives and Negatives, based on the confidence score (or probability assignment) a classification model yields for each label. Let's start by defining some terminology to establish a formal definition. Consider:

1. A dataset $S : (\mathbf{x_1}, y_1), ..., (\mathbf{x_n}, y_n) \in \mathbb{R}^p \times \{C_1, ..., C_m\}$, where
   - $\mathbf{x_i}$ is a vector of p features corresponding to sample i
   - y_i is the class corresponding to sample i
   - $\{C_1, ..., C_m\}$ is the set of possible classes

2. A classification model $M : \mathbb{R}^p \mapsto \{C_1, ..., C_m\}$ trained to predict label assignment given an input vector $\mathbf{x_i}$. The model assigns a *confidence score* (or *probability* if the model is probabilistically calibrated) to each possible class C_j for any given input vector $\mathbf{x_i}$, signifying the model's *confidence* that C_j is the true class for the given input vector (which can also be expressed as $C_j = y_i$). Let's call this confidence score $M(\mathbf{x_i}, C_j)$. The class with the highest confidence score will be the model's predicted class $\widehat{y_i}$.

We have, for any sample $i \in \{1, ..., n\}$:

$$\sum_{j=1}^{m} M(\mathbf{x_i}, C_j) = 1 \tag{1}$$

$$\widehat{y_i} = \arg\max_j (M(\mathbf{x_i}, C_j)) \tag{2}$$

By applying the model M on the full dataset S, we obtain a confidence score $M(\mathbf{x_i}, C_j)$ for each $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$. Suppose S_j denotes the set of samples with true class C_j. We can build a probabilistic confusion matrix pCM as follows:

$$pCM(j_{ref}, j_{hyp}) = \sum_{i \in S_{j_{ref}}} M(\mathbf{x_i}, C_{j_{hyp}}) \tag{3}$$

Intuitively, each cell (j_{ref}, j_{hyp}) of the confusion matrix corresponds to the total confidence score assigned by the model to hypothesis j_{hyp} for samples for which the true class is j_{ref}. It is very similar to the usual definition of a confusion matrix, apart from the fact that we leverage all confidence scores as quantitative values as rather than just the highest-scoring class as a qualitative value. From this probabilistic confusion matrix, cRecall and cPrecision are calculated in the same way that Recall and Precision are from the non-probabilistic (regular) confusion matrix.

We can also formulate it without using a confusion matrix, by using indicator functions. The commonly used definition of *true positive for class* C_j is any model prediction for which $\widehat{y_i} = y_i = C_j$. We can formalize it as:

$$TP_{C_j} = I_{\widehat{y_i}=C_j} * I_{C_j=y_i} \tag{4}$$

$$\text{where: } I_X = \begin{cases} 1 \text{ if } X \text{ is true} \\ 0 \text{ if } X \text{ is false} \end{cases}$$

We propose a continuous generalization as the *confidence true positive*:

$$cTP_{C_j} = M(\mathbf{x_i}, C_j) * I_{C_j=y_i} \tag{5}$$

As shown in Equation 5, we're simply replacing the binary $I_{\widehat{y_i}=C_j}$ from Equation 4 by the continuous $M(\mathbf{x_i}, C_j)$. We can similarly define the *confidence False Positive* as

$$cFP_{C_j} = M(\mathbf{x_i}, C_j) * I_{C_j \neq y_i} \tag{6}$$

Now that we formalized cTP and cFP, we can define $cPrecision$ and $cRecall$:

$$cPrecision = \frac{cTP}{cTP + cFP} \tag{7}$$

$$cRecall = \frac{cTP}{TP + FN} \tag{8}$$

Note the asymmetry between cPrecision and cRecall, as the denominator of cRecall is the same as the denominator of Recall (does not use the probabilistic extensions of FP and FN). This is because $TP + FN$ simply refers to the total number of samples labelled as the class being evaluated.

3 Related Work

In the publication Employ Decision Values for Soft-Classifier Evaluation with Crispy References, Zhu et al. (2018) have come to a similar formulation of probabilistic confusion matrix in the pursuit of a different goal. Zhu considered the use-case of soft-classification, where "the classifier outputs not only a crispy prediction about its class label, but decision values which indicate to what extent does it belong to the all the classes as well", while we're considering hard classification, where the hypothesis probabilities output by the classifier indicate a confidence score that the hypothesis is correct, rather than a measure of class membership. From the resulting confusion matrix, Zhu also formulated and empirically experimented with a probabilistic version of Precision and Recall, but only for binary classification. In our paper we dive deeper into the properties and potential of these metrics in multiclass hard classification when the model hypothesis confidence scores are impactful to the use case, especially in large-scale NLP systems.

Many publications (Dinga et al., 2019; Hossin M, 2015) have shed light on pitfalls of commonly used evaluation metrics, and introduced alternatives and best practices to avoid those pitfalls. However, the criticized metrics have maintained their status as the standard in most industries and in academia.

Ling et al. (2003) and Vanderlooy and Hüllermeier (2008) have proposed methodologies to evaluate metrics against each other. We decided however to approach this problem from a different perspective. We will only compare a metric to its proposed extended counterpart (e.g. F1 vs cF1), and will not claim our proposed metrics to be objectively better, but simply demonstrate advantages they introduce, and in which situations those advantages are useful. In many use-cases it might still be preferable to use the regular Precision and Recall.

There are many existing metrics that leverage the model's probability assignments over classes. Brier score (Brier, 1950) is an example, and is widely accepted as a standard in probabilistic weather forecasting. It is a strictly proper scoring rule, meaning it is uniquely optimized by reporting probabilistically calibrated model predictions. Using our earlier defined methodology, Brier-score can be calculated as: $BS = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}(M(\mathbf{x_i}, C_j) - I_{C_j=y_i})^2$

which can be interpreted as a sum of squared errors between the predicted probability distribution and the true distribution. Brier-score is effective at giving a big picture of model performance beyond the top model hypothesis, along with an evaluation of probabilistic calibration.

However, Brier-score is not appropriate for large-scale NLP systems such as those described in Section 1, for two main reasons. The first one is that the data distribution is often imbalanced, as basic commands such as "stop" or "play" are dominating as compared to more niche features such as "open halo on my xbox", while the importance of a class is not reflected in its distribution (e.g. calling for emergency). This shows that these use cases require class-based measures, rather than aggregated ones like Brier-score. Secondly, it is important to be able to evaluate each class independently to understand the class tradeoffs (False Accepts and False Rejections from/towards competing classes), as different stakeholders are responsible for different functionalities. Additionally, Brier-score can be difficult to interpret and explain to non-technical stakeholders, as compared to other common metrics such as Precision and Recall. The concerns presented in this paragraph also hold for other probabilistic metrics we found in literature, such as the Probabilistic Confusion Entropy (Wang et al., 2013) and metrics usually used as loss functions during training.

Another widely popular metric is the Area Under the Receiver Operating Characteristic Curve (AUC), which is originally for binary classifiers, but has been generalized to handle multi-class (Hand and Till, 2001). AUC is taxonomized as a rank-based metric rather than a probabilistic metric, as it is only sensitive to changes in confidence scores when those changes cause a difference in the ranking of test samples. Multiple extensions of AUC have been proposed to allow it to better leverage probability score assignments, such as the pAUC (Ferri et al., 2004) and soft-AUC (Calders and Jaroszewicz, 2007) but these extensions were only defined and analyzed in the binary classification case, and were also questioned by Vanderlooy and Hüllermeier (2008) through empirical experiments indicating that the variants fail to be more effective than the original AUC.

4 Benefit A: Robustness to NaN values

Precision, Recall, and F1-Score have a denominator which in some cases can be equal to zero, making it impossible to calculate an estimate of the metric. In the case of Recall, this denominator (TP + FN) would be equal to zero for any label that is not present in the test-set (no sample in the test-set is assigned this label as its ground truth). As Recall and cRecall have the same denominator, this situation would also cause cRecall to be NaN for those labels. Precision would be NaN for any label that is not hypothesized by the model when making predictions in the test-set.

Notably, cPrecision does not fall victim to the same issue (except in the extreme case when the model never assigns any confidence to that label). This is due to the fact that cPrecision considers that the model is making soft predictions (confidence score assignments) for all labels for each test sample. This quality makes cPrecision more robust than Precision. The number of NaN values for Precision increases as the test-set size goes smaller, which also affects F1, as a NaN Precision causes a NaN F1. This makes cF1 less likely to have NaN values than F1 under small test-set sizes. Especially in imbalanced datasets or datasets with a large number of possible classes, this issue is burdensome as your test-set needs to be very large to have a high likelihood of all classes being hypothesized.

5 Benefit B: Model confidence score sensitivity

Intuitively, it appears as a good characteristic for a model to be more confident when being correct and less confident when being wrong. Even though in some use cases this might not result in better outcomes for the stakeholders since only the highest-score prediction is actually used in the final application. Nonetheless, selecting a better model can yield benefits in the long term, or when noise/out-of-sample data is introduced, as the score assignments reflect how well the model understands the underlying data. We did not deem it necessary to perform an empirical demonstration of this quality, as it is a central aspect in the definition of the proposed metrics.

In order to concretely illustrate the practical use of Benefit B in an NLP production environment, consider a case where a company is building a semi-automated text labelling pipeline, where a model automatically labels text samples when the top prediction's confidence score is higher than a chosen threshold (e.g. 0.99), and sends the remaining samples to humans for manual labelling (e.g. Amazon SageMaker Ground Truth as an NLP pipeline). In cases where the label space is large (common in large-scale NLP) human annotators cannot be familiar with all annotations. To address this, the pipeline presents the human annotator with the model's n-best output predictions as suggestions, to improve efficiency and reduce their burden. Evaluating this NLP model with threshold-based metrics would not be appropriate, as the full n-best output is used to influence the human annotator's decisions. Metrics such as Brier-score and Log-loss would be a step forward, but would not allow for a balanced class-based evaluation, and would not give visibility over the tradeoffs between classes. The latter metrics are also difficult to interpret, as the stakeholders are likely to be interested in measures they can easily relate to, and a potential break-down of which labels are more difficult to identify. In such case, cPrecision, cRecall, and cF1 would be appropriate, as they would bring the thorough evaluation of probabilistic metrics combined with the interpretability and robustness to class imbalance of Precision and Recall.

6 Empirical Results

In our experiments we directly compare each of the 3 threshold-based metrics with their probabilistic counterpart. Our goal is to support the claimed benefits of *generalization* and *functional consistency*. We present results on the SNLI dataset (Bowman et al., 2015), a dataset of paired sentences (560152 training samples and 10000 test samples) rated by annotators as either “Neutral”, “Entailment”, or “Contradiction” depending on how the sentences relate to each other. Please note that as SNLI does not suffer from data imbalance and only has 3 classes, this experiment is not intended to illustrate all of the advantages of the proposed metrics in large-scale commercial NLP pipelines. Instead, we simply use SNLI to experimentally support the *generalization* and *functional consistency* benefits.

6.1 Testing Hypothesis C: Generalization

We trained Sep-CNN (Denk, 2018) models (structure shown in Figure 7 in Appendix B) with different sets of randomly chosen hyperparameters and training samples. Sep-CNN was used here for its training and evaluation speed that eased speed of experimentation. More replication information can be found in Appendix A.

Generalization is directly tied to the variance of the metrics. Metrics with lower variance will have tighter confidence bounds, which implies that their point estimates are closer to the true population values. This means that a point estimate calculated from a small dataset is more likely to be generalizable to unseen data. Lower variance also implies that less data is required to reach statistically significant results in discriminating models. For this hypothesis we simply need to demonstrate that our proposed metrics have lower variance than their threshold-based counterpart. We set-up our experiments also to show that our proposed metrics are better able to discriminate between models.

We introduce differences in the models by varying the percentage of the training dataset selected during subsampling for model training, and injecting noise by changing a certain percentage of the labels on that data to a random alternative label. Model 1 used 100% of the training data, Model 2 used 66.7% of the training data, 10% of which is altered to introduce noise, and finally Model 3 used 33.3% of the training data, 20% of which is altered to introduce noise. The goal of changing these two parameters is to create enough differentiation in performance between the models in order to have preliminary expectations of which models will perform best. Model 1 is expected to perform better than Model 2, which is expected to perform better than Model 3.

We then ran predictions on test-set down-samplings of 1.0, 0.5, 0.2, 0.1, 0.05, 0.02, and 0.01 ratios for each of these models and used the bootstrap method (Efron, 1979) with 1000 resamplings, to calculate the mean and 95% confidence intervals for the F1 and cF1 scores for each class, for each model, on each test data down-sampling. Figure 1 below presents a comparison of the resulting variances of cF1 and F1. Figures 5 and 6 (Appendix B) show the same comparison but for cPrecision against Precision, and cRecall against Recall respectively.

The plots show two key elements: variances get smaller as the test-set size increases, and the variance of the probabilistic metrics is always lower than the variance of their threshold-based counterpart. We also used a F-test of equality of variance, Bartlett’s test, and Levene’s test to reject the null hypothesis that the variance of the threshold-based metric and its probabilistic counterpart are equal, and obtained statistically significant results ($pvalues < 0.05$) in all cases.

In Figure 2 we compare F1 and cF1’s abilities to discriminate between models, at different test-set sizes. The shaded region for each line represents the 95% confidence interval. The x-axis represents down-sampling ratios of the test-set used for each metric evaluation. We see the confidence intervals being further away from each other for cF1 as compared to F1, across all test-set sampling sizes, allowing for a statistically significant identification of which models have a better understanding of the underlying data. Figures 3 and 4 in Appendix B

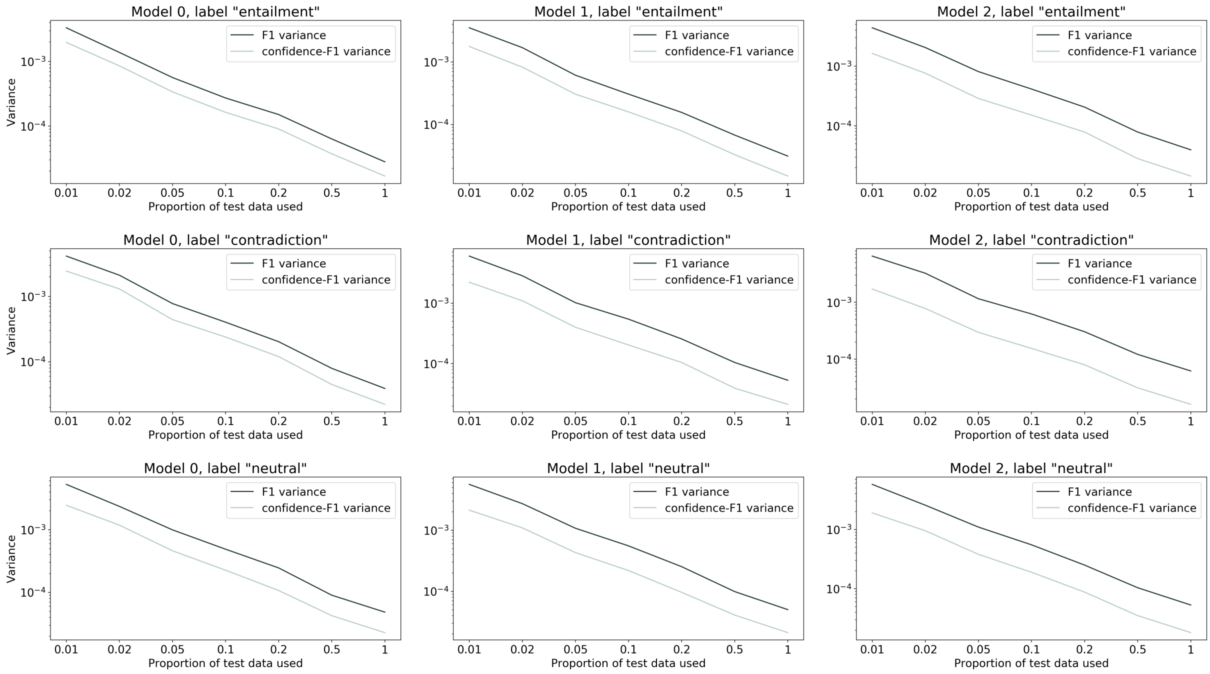

Figure 1: Empirical comparison of the variances of F1 and cF1, across different test-set sizes.

show the same but for Precision against cPrecision and Recall against cRecall respectively.

6.2 Testing Hypothesis D: Functional Consistency

We hypothesize that for most pragmatic and real-world modeling decisions, the ranking of performance of candidate models by each metric (when compared to their threshold-based counterpart) is the same when tested on a test-set similar enough to the population distribution of test data. Figure 2 illustrates it, by showing consistency in model rankings for each metric at the 1.0 subsampling (full test-set size).

In order to more empirically demonstrate this observation, we ran an experiment where we randomly generated 100 models by sampling from a selection of different possible hyperparameters. We then compared all models against each other, resulting in 4950 pairwise comparisons, using the 6 metrics considered (Precision, cPrecision, Recall, cRecall, F1, cF1). From these results, we extracted all the cases in which both the probabilistic and the thresholded metric showed a statistically significant difference between the two models being compared (t-test with p=0.01). Among the latter cases, we counted the percentage of agreement (cases where both metrics agree on which model is better). The results for each class are demonstrated in Table 1. These results indicate that comparable metrics (i.e. Precision and cPrecision, Recall and cRecall, F1 and cF1) agree the majority of the time.

Metric Type	Class	% Sig.	% Agree
(c)F1	entailment	93.29	89.19
(c)F1	contradiction	93.61	93.75
(c)F1	neutral	90.32	84.94
(c)Precision	entailment	92.97	94.52
(c)Precision	contradiction	94.20	92.62
(c)Precision	neutral	89.78	80.11
(c)Recall	entailment	95.84	75.81
(c)Recall	contradiction	95.75	84.99
(c)Recall	neutral	91.71	81.13

Table 1: Percent of statistically significant model comparisons that agree between each pair of comparable metrics.

7 Potential Shortcomings to Consider

As mentioned in the introduction, the proposed metrics do not evaluate for probabilistic calibration. There are many cases in which model probability scores are used in production with the expectation of reflecting reliable probabilities. In such cases, probabilistic calibration would have to be evaluated separately using a strictly proper scoring rule (Gneiting and Raftery, 2007).

Another aspect to consider is interpretability.

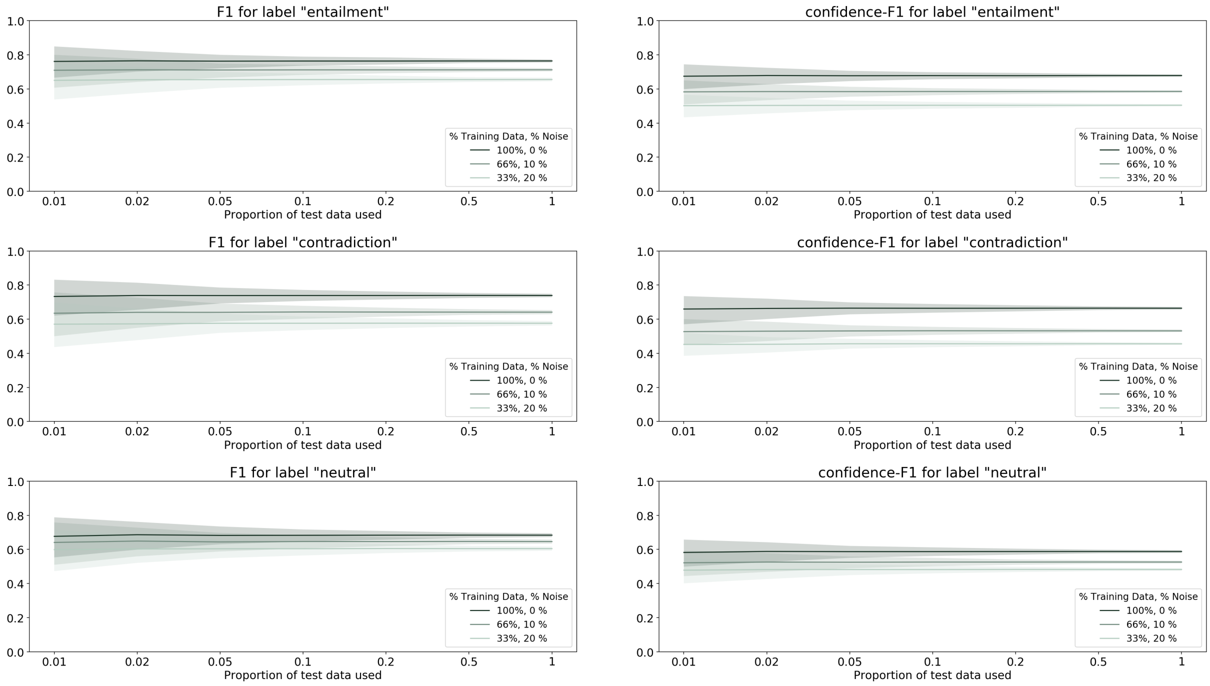

Figure 2: Empirical comparison between F1 and cF1 scores, across different levels of train-set sampling and noise, and different levels of test-set sampling. The y-axis represents the F1 and confidence-F1 values.

The proposed extensions lack the degree of direct interpretability afforded by their threshold-based counterparts. We believe that they still have a high degree of interpretability when compared to other metrics such as Brier Score and Cross-Entropy.

We believe that these downsides do not necessarily pose a problem as long as they are known to the users of the metrics, so they can take appropriate measures in cases when it is required.

8 Future Work

In this paper we only focused on classification, and not named entity recognition (NER), while NLP often requires both. Many NER specific metrics like SER (Makhoul et al., 1999) consider the possibility of having slot insertions or deletions, making them more appropriate for evaluating NER. In the future we hope to extend metrics like SER to gain these benefits.

We also hope to investigate the feasibility of altering these proposed metrics to be Strictly Proper Scoring Rules (Gneiting and Raftery, 2007) allowing for a dual assessment of probabilistic calibration and performance. Strict Proper Scoring will aid us as we plan to study the potential use of these metrics as a differentiable model loss for training.

Finally, we hope to soon address the question of how to deal with output quantization where discrete confidence bins (HIGH, MED, LOW) rather than the continuous values are used by downstream tasks or customers.

9 Conclusion

We introduced probabilistic extensions of widely used threshold-based metrics, and four benefits they provide as compared to their original counterparts. These benefits motivate the use of our proposed metrics in real-world problems where data is scarce and/or where the model confidence score assignments over its predictions are leveraged in production. We hope these metrics will allow for more reliable modeling decision-making in such cases. We hope this research will pave the way for further investigation into the challenge of model evaluation with under-representative test-sets.

Acknowledgments

The authors would like to thank Sreekar Bhaviripudi, Jack FitzGerald, Spyros Matsoukas, and Cedric Warny for reviewing this work and providing valuable feedback. The authors would also like to thank the anonymous reviewers for their insightful comments and suggestions.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Glenn W. Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3.

Toon Calders and Szymon Jaroszewicz. 2007. Efficient auc optimization for classification. In *Knowledge Discovery in Databases: PKDD 2007*, pages 42–53, Berlin, Heidelberg. Springer Berlin Heidelberg.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Timo Denk. 2018. Text classification with separable convolutional neural networks.

Richard Dinga, Brenda W.J.H. Penninx, Dick J. Veltman, Lianne Schmaal, and Andre F. Marquand. 2019. Beyond accuracy: Measures for assessing machine learning models, pitfalls and guidelines. *bioRxiv*.

B. Efron. 1979. Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, 7(1):1–26.

C. Ferri, J. Hernández-Orallo, and R. Modroiu. 2009. An experimental comparison of performance measures for classification. *Pattern Recogn. Lett.*, 30(1):27–38.

Cèsar Ferri, Peter Flach, José Hernández-orallo, and Athmane Senad. 2004. Modifying roc curves to incorporate predicted probabilities. In *In Second Workshop on ROC Analysis in ML*.

Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org.

David J. Hand and Robert J. Till. 2001. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186.

José Hernández-Orallo, Peter Flach, and Cèsar Ferri. 2012. A unified view of performance metrics: Translating threshold choice into expected classification loss. *J. Mach. Learn. Res.*, 13(1):2813–2869.

Sulaiman M.N Hossin M. 2015. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1–11.

Charles X. Ling, Jin Huang, and Harry Zhang. 2003. Auc: A statistically consistent and more discriminating measure than accuracy. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, page 519–524, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

John Makhoul, Francis Kubala, Richard Schwartz, and Ralph Weischedel. 1999. Performance measures for information extraction. In *In Proceedings of DARPA Broadcast News Workshop*, pages 249–252.

Allan H. Murphy. 1973. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4):595–600.

S. J. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

R. Sarikaya. 2017. The technology behind personal digital assistants: An overview of the system architecture and key components. *IEEE Signal Processing Magazine*, 34(1):67–81.

Stijn Vanderlooy and Eyke Hüllermeier. 2008. A critical analysis of variants of the auc. *Machine Learning*, 72(3):247–262.

Xiao-Ning Wang, Jin-Mao Wei, Han Jin, Gang Yu, and Hai-Wei Zhang. 2013. Probabilistic confusion entropy for evaluating classifiers. *Entropy*, 15(12):4969–4992.

Lei Zhu, Tao Ban, Takeshi Takahashi, and Daisuke Inoue. 2018. Employ decision values for soft-classifier evaluation with crispy references. In *Neural Information Processing*, pages 392–402, Cham. Springer International Publishing.

A Reproducibility Information

A.1 Preprocessing

All sentences were lowercased, periods were stripped from the ends of sentences. When found in the middle of sentences the periods are space-separated so that they are separate tokens. Sentence pairs were separated with the ”[SEP]” token. Tokens were given indices up to the 20000th token, after which tokens are assigned to a reserved index indicating OOV. A max sequence length of 42 was chosen for speed, based on the distribution of lengths in the SNLI dataset. Samples longer than 42 were truncated.

A.2 Model Training

All models were trained with:

- 150 epochs
- Early stopping on validation loss with 8 epochs of patience
- Randomized validation split with 9:1 train to validation ratio
- Batch size 128
- Learning rate 1e-3
- Adam optimizer
- Loss: sparse categorical cross-entropy
- 1d max pooling (pool size=1) between each convolution layer and average pooling before the dense layer output

Hyperparameter	Possible Values
Blocks	1, 2
Filters	32, 64, 128, 256
Kernel Size	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Embedding Dim.	64, 128, 256
Dropout rate	0.0, 0.1, 0.2, 0.3, 0.4, 0.5

Table 2: Table of Hyperparameter choices.

When generating and training the 100 models for the testing of Hypothesis D, we randomly drew hyperparameters from the following distributions, shown in Table 2, with no two models sharing the same hyperparameters (checked for redundancy).

B Additional Figures

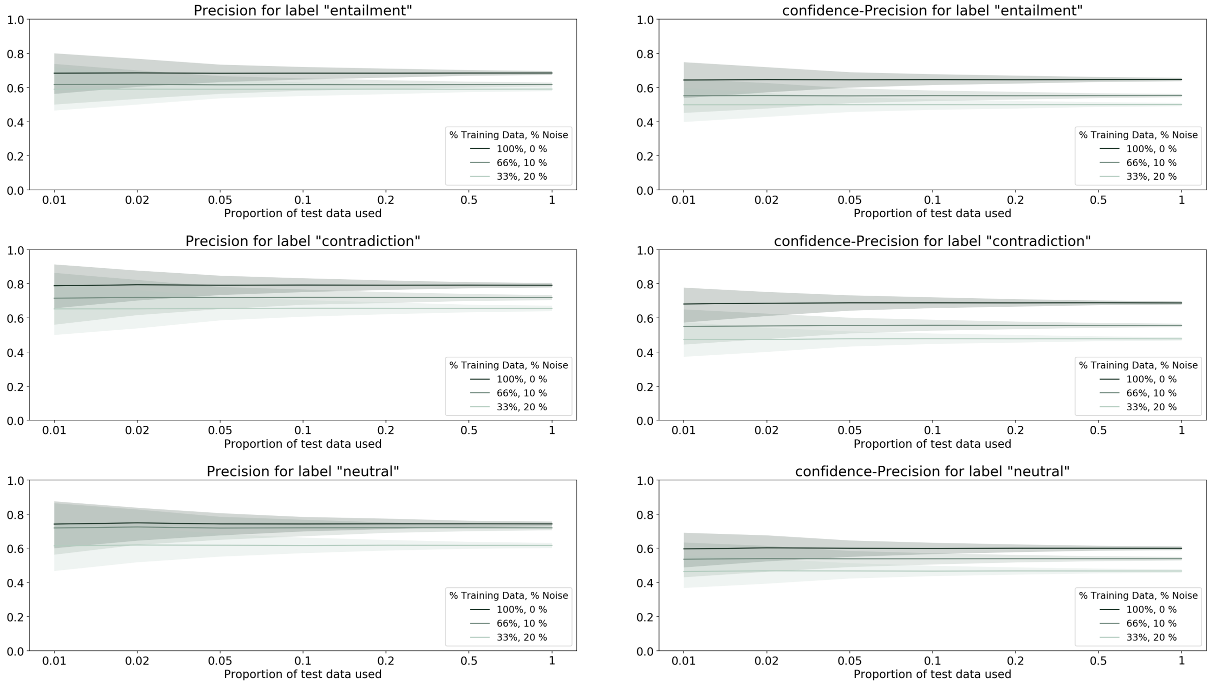

Figure 3: Empirical comparison between Precision and cPrecision scores, across different levels of train-set sampling and noise, and different levels of test-set sampling. The y-axis represents the Precision and confidence-Precision values.

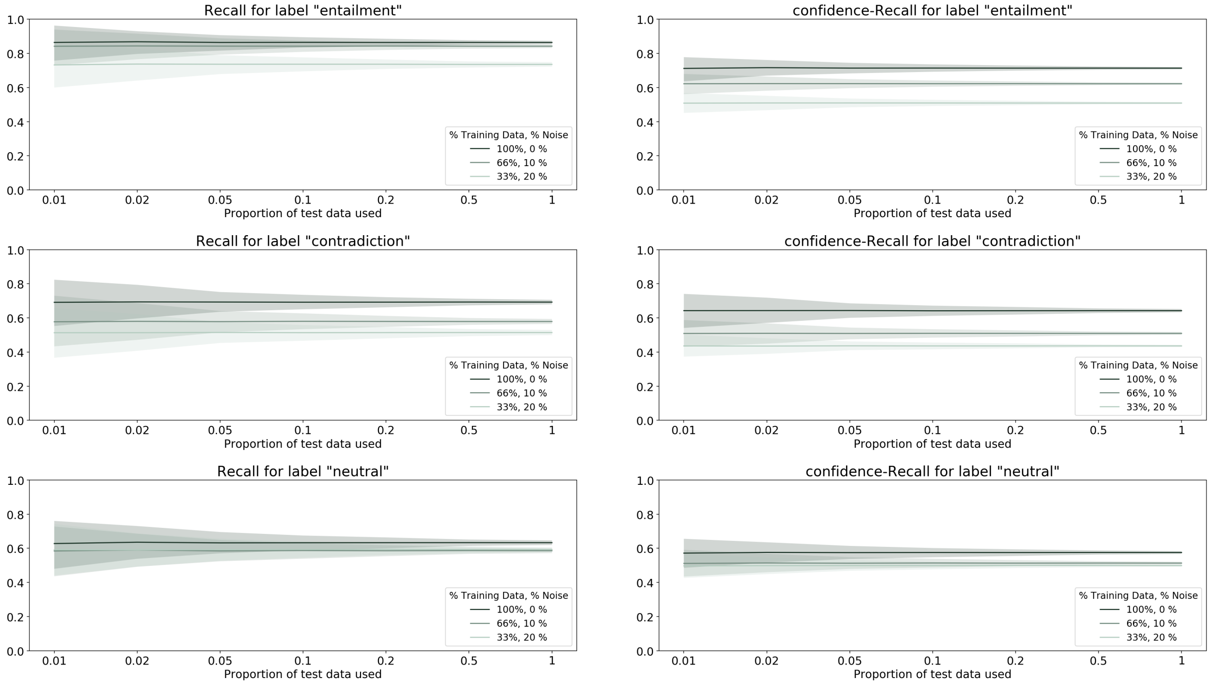

Figure 4: Empirical comparison between Recall and cRecall scores, across different levels of train-set sampling and noise, and different levels of test-set sampling. The y-axis represents the Recall and confidence-Recall values.

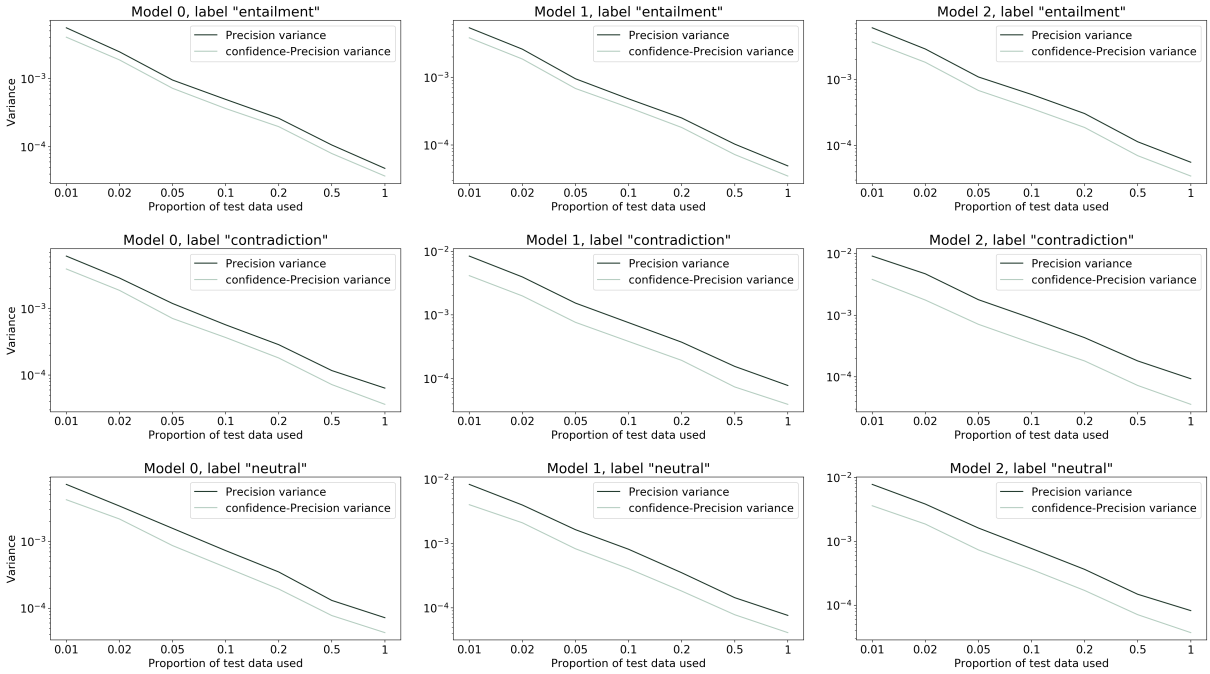

Figure 5: Empirical comparison of the variances of Precision and cPrecision, across different test-set sizes

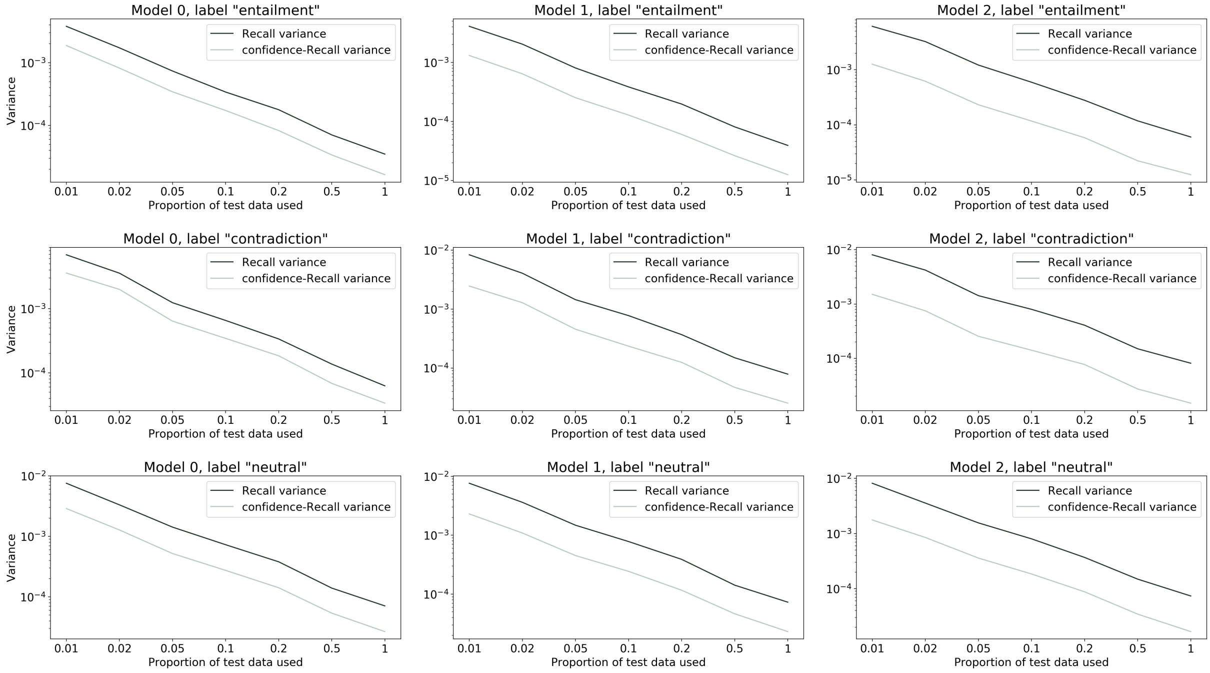

Figure 6: Empirical comparison of the variances of Recall and cRecall, across different test-set sizes

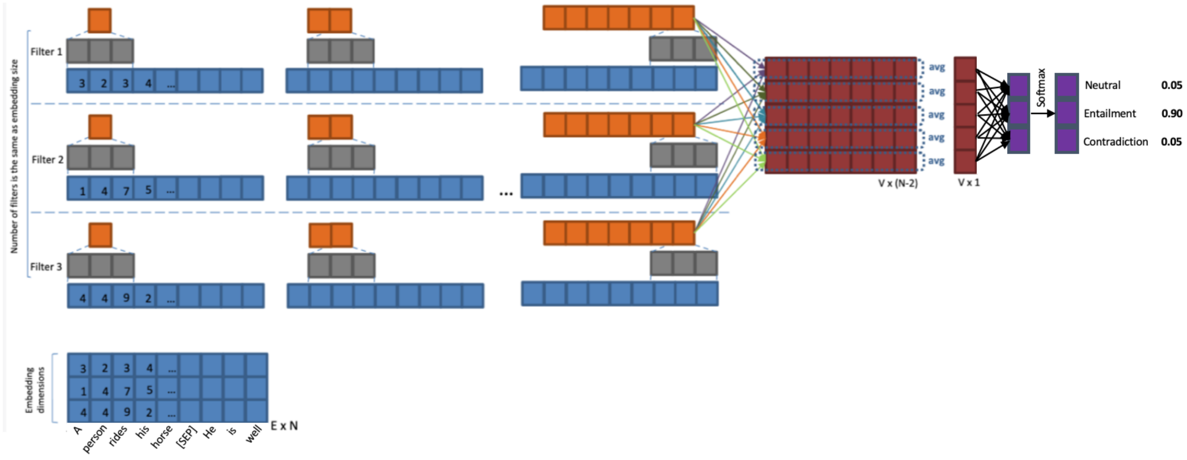

Figure 7: Sep-CNN model architecture used for experimentation on the NLP dataset