WRF: Weighted Rouge-F1 Metric for Entity Recognition

Lukas Jonathan Weber [◇] **Krishnan Jothi Ramalingam**[◇,*]

Matthias Beyer[◇] **Axel Zimmermann**[†]

[◇] Mercedes-Benz AG, Stuttgart
[†] Steinbeis-Transferzentrum (esz)
{lukas.l.weber, krishnan.jothi_ramalingam, matthias.beyer}@mercedes-benz.com
zimmermann@enseg.de

Abstract

The continuous progress in Named Entity Recognition allows the identification of complex entities in multiple domains. The traditionally used metrics like precision, recall, and F1-score can only reflect the classification quality of the underlying NER model to a limited extent. Existing metrics do not distinguish between a non-recognition of an entity and a misclassification of an entity. Additionally, the dealing with redundant entities remains unaddressed. We propose WRF, a **W**eighted **R**ouge **F**1 metric for Entity Recognition, to solve the mentioned gaps in currently available metrics. We successfully employ the WRF metric for automotive entity recognition, followed by a comprehensive qualitative and quantitative analysis of the obtained results.

1 Introduction

The continuous progress in Named Entity Recognition (NER) allows the identification of complex entities in multiple domains (Sharma et al., 2022). The traditionally used metrics like precision, recall, and F1-score (Tjong Kim Sang and De Meulder, 2003) can only reflect the classification quality of the underlying NER model to a limited extent (Powers, 2015). The limitation of the entity recognition evaluation metrics has been studied by many researchers, which motivated them to modify the existing or create new metrics (ACE08, 2008; Chinchor and Sundheim, 1993; Segura-Bedmar et al., 2013) to tackle many corner cases (Ben Jannet et al., 2014). This research work shows that still many corner cases are not being addressed by the existing metrics to date, to evaluate the true prediction performance of the model. In the NER task, the model needs to identify the entity and classify it. After tokenizing the input text, all the tokens that do not represent an entity of our interest are usually labeled as *other* (O). Existing metrics do not distinguish between a non-recognition of an entity and a misclassification of an entity. Non-recognition is the wrong classification of an entity as *other*, whereas a misclassification is the wrong classification of an entity as any of the other classes, apart from *other*. Furthermore, the dealing of redundant entities which are present in the predicted or target labels are not tackled by the above-mentioned metrics and therefore should take into account too.

In this work, we show that the existing metrics do not fit well for Automotive Entity Recognition (AER). AER is the automotive domain-specific entity recognition task. We propose WRF, a **W**eighted **R**ouge **F**1 metric for Entity Recognition, to solve the gaps in currently available metrics. The scientific contribution is structured as follows: In Section 2, we give insights into related work. The currently available metrics, the identified metric gap, and our proposed method WRF are explained in Section 3. Section 4 deals with the fine-tuning of a pretrained language model with an AER dataset and the quantitative and qualitative evaluation comparison between existing metric and WRF. We will end up this contribution with a conclusion in Section 5.

2 Related work

The evaluation of entity recognition models is a crucial task in the field of NLP. Several forums have addressed meaningful entity recognition evaluation metrics in the past. The entity recognition challenge (Tjong Kim Sang and De Meulder, 2003) at the conference on computational natural language learning 2003 (CoNLL2003) introduced the idea of measuring the performance of the systems in terms of precision, recall, F1, and its variations like F1-micro, which considers the entity prediction to be correct, only when the sequence of predicted labels for the entire entity precisely matches the sequence

[*] Work done during an internship at Mercedes-Benz AG.

Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems, pages 1–11

Prediction: Repair costs (parts and labour) are often very high, since the workshop does not know which is the faulty $^{\text{FP}}$ location and then also replaces the ekmv $^{\text{FP}}$ or replaces because of the consequential damage to the scroll $^{\text{FP}}$, (scroll tip $^{\text{TP}}$ is partially melted $^{\text{TP}}$) by too high temperatures $^{\text{TP}}$.	
Target: Repair costs (parts and labour) are often very high, since the workshop does not know which is the faulty location and then also replaces the ekmv or replaces because of the consequential damage to the scroll, (scroll tip is partially melted) by too high temperatures.	
-	**Failure location**	**Failure type**
True Positives (TP)	1 (scroll tip)	2 (partially melted, too high temperatures)
False Positives (FP)	2 (ekmv, scroll)	1 (faulty)
False Negatives (FN)	-	-
Recall	1/(1+0) = 1.00	2/(2+0) = 1.00
Precision	1/(1+2) = 0.33	2/(2+1) = 0.67
F1-Score	0.50	0.80
F1 micro	**0.67 (TP=3, FP=3, FN=0)**	

Table 1: A practical use-case for F1-score calculation. The target describes the gold annotated labels by humans. Recall, Precision, and F1-Score are computed based on the target and prediction entities. The metrics were calculated separately for **failure location** and **failure type**. Underlined entities are defined as the beginning of an entity sequence.

of true labels, token by token (Tjong Kim Sang and De Meulder, 2003). In other words, there is no room for variation or flexibility in the sequence of tokens used to represent the entity in the predicted label and the true label. F1 metric and its variants are widely used in the entity recognition field (Yadav and Bethard, 2018). The automatic content extraction (ACE08, 2008) research program provided three additional metrics for evaluating entity recognition tasks, which are defined as entity scoring, relation scoring, and event scoring. Chinchor and Sundheim (1993) defined different classification categories such as partial and spurious, to compare the response of a system against the target annotation. Partial is defined as the predicted entity and the target entity is judged to be a near match, whereas spurious is the hypothesising of an entity by the model. They build up a new metric called error per response fill, based on their classification categories. The idea is to go beyond simple strict classification and provide flexibility in evaluation. Building upon the categories defined by Chinchor and Sundheim (1993), Segura-Bedmar et al. (2013) created four schemes to provide more flexible evaluations, namely strict evaluation, exact boundary matching, partial boundary matching, and type matching, which solve a wider range of use cases displayed in Section 3.2. Fu et al. (2020) introduced an interpretable evaluation method for entity recognition tasks. The method offers possible insights into the underlying reasons behind the differences between the performances of the models, which is not attainable through conventional metrics.

3 Method

3.1 Automotive Entity Recognition

AER deals with the identification of failure locations and failure types in unstructured customer feedback texts in the automotive warranty and goodwill area (W&G). In the automotive industry, these identified entities are used to eliminate frequent failures and improve product quality. We display automotive W&G text for visualization purposes. The following sentence was classified with a BERT-base uncased (Devlin et al., 2019) token classification model, which was fine-tuned with an AER-labeled dataset (details for training in Section 4.1).

Repair costs (parts and labour) are often very high, since the workshop does not know which is the faulty location and then also replaces the ekmv or replaces because of the consequential damage to the scroll, (scroll tip is partially melted) by too high temperatures.

The automotive entity classifications based on the fine-tuned BERT-model are displayed in the following example sentence.

	Prediction 1: Repair costs (parts and labour) are often very high, since the workshop does not know which is the faulty location and then also replaces the ekmv ^{FP} or replaces because of the consequential damage to the scroll ^{FP}, (scroll tip ^{TP} is partially melted) by too high temperatures.	**Prediction 2**: Repair costs (parts and labour) are often very high, since the workshop does not know which is the faulty location and then also replaces the ekmv ^{FP} or replaces because of the consequential damage to the scroll, (scroll tip ^{TP} is partially melted) by too high temperatures.
Calculation	Recall: 1.00 Precision: 0.33 F1-score: 0.50	Recall: 1.00 Precision: 0.50 F1-score: 0.67
Problem	The classification result of scroll (multiple occurrences) should not affect the evaluation metric, since it neither conveys any useful information nor any wrong information. The repetitive and redundant entities influence the F1.	

Table 2: The problematic use-case for F1-score calculation. We display the calculation of the entity failure location for simplicity reasons.

Repair costs (parts and labour) are often very high, since the workshop does not know which is the faulty ^{FP} location and then also replaces the ekmv ^{FP} or replaces because of the consequential damage to the scroll ^{FP}, (scroll tip ^{TP} is partially melted ^{TP}) by too high temperatures ^{TP}.

3.2 NER evaluation schemas

The use cases which can be dealt with by CoNLL2003 metrics are displayed in Table 7. Use cases which can be exclusivly handled with the SemEval'13 metrics are displayed Table 8.

The use case in Table 9 is missing according to our investigation (Section 3.1). Redundant entities do not contribute to the failure elimination process in the automotive industry, and result in an imprecise calculation of the F1 score. To illustrate the problem of calculating the F1-Score, we take the example sentence from Section 3.1 for calculating the metrics precision, recall, and F1-Score. The metrics calculation is done in Table 1. The problem of using the F1-Score metric is shown in Table 2.

3.3 WRF: Weighted Rouge-F1 metric for Entity Recognition

Rouge score (Lin, 2004) is commonly used in text-generation tasks to compare the model-generated text against the reference or a set of human-generated reference texts (Schluter, 2017). It has several variants, such as Rouge-N, Rouge-L, and Rouge-W. Our interest is centered on the Rouge-N variation, specifically in the unigram version, the Rouge-1 F1 (R1-F1). For our particular use case, there is no necessity to match lengthier sequences of multiple words or n-grams because the majority of the entities associated with failure location and types are single words or unigrams. Since this research deals with the classification task, the first step is to create two texts from the predicted and target entities, to compare and evaluate the quality of predictions using the rouge score. The need to adapt a commonly used text-generation evaluation metric for the classification task and how it will be beneficial will be made clear before the end of this section. The example described in subsection 3.1 is used to evaluate the failure location and failure type predictions using the R1-F1 in Table 3.

$$\text{R1-Precision} = \frac{count_{match}(gram_1)}{count(gram_1)_{model}} \quad (1)$$

$$\text{R1-Recall} = \frac{count_{match}(gram_1)}{count(gram_1)_{reference}} \quad (2)$$

$$\text{R1-F1} = 2 \times \frac{Precision_{RP1} \times Recall_{RR1}}{Precision_{RP1} + Recall_{RR1}} \quad (3)$$

where $count_{match}(gram_1)$ refers to the number of unigram matches found between the model prediction and the reference, $count(gram_1)_{model}$ refers to the number of unigrams in the model prediction, and $count(gram_1)_{reference}$ refers to the number of unigrams in the reference. The initial step in evaluating entity recognition performance using R1-F1 is to construct two strings, P and T, using the predicted and target entities. The string P is the concatenation of predicted entities, whereas the string T is the concatenation of target entities. M_c determines the number of unigrams that P and T have in common. R1-Precision is calculated as the ratio of M_c to P_c, where P_c is the total number of unigrams in P. R1-Recall is calculated as the ratio of M_c to T_c, where T_c is the total number of unigrams in T. R1-F1 is calculated as the harmonic mean of recall and precision. The repetitive words are taken into account during the computation of P_c.

A new evaluation metric called Weighted Rouge

Prediction: Repair costs (parts and labour) are often very high, since the workshop does not know which is the faulty ^{FP} location and then also replaces the ekmv ^{FP} or replaces because of the consequential damage to the scroll ^{FP}, (scroll tip ^{TP} is partially melted ^{TP}) by too high temperatures ^{TP}.		
Rouge-1 F1-score (unigram)		
Form string T from target entities	scroll tip partially melted too high temperatures	
Form string P from predicted entities	faulty ekmv scroll scroll tip partially melted too high temperatures	
$\mathbf{M_c}$: No. of unigram (word) matches between P & T	**7** (scroll tip partially melted too high temperatures)	
$\mathbf{P_c}$: No. of word in P	**10**	
$\mathbf{T_c}$: No. of word in T	**7**	
Calculation	R1-Precision = M_c/P_c = 7/10 = 0.70 R1-Recall = M_c/T_c = 7/7 = 1.00 R1-F1-score = 0.82	Rouge-1 Precision (Equation 1) Rouge-1 Recall (Equation 2) Rouge-1 F1 (Equation 3)
Conclusion	To measure the prediction performance of the AER-specific W&G-BERT model, we choose an modified Rouge-1 F1-Score (WRF: Weighted Rouge F1 metric for Entity Recognition).	

Table 3: Rouge-1 F1-score calculation.

F1 (WRF) is introduced in Table 4. Since we are interested in the unigram matching, it is weighed R1-F1[1], to mitigate the issues described in subsection 3.2 and Table 2. This is a modified version of R1-F1 for entity recognition from Table 3. The example from subsection 3.1 is taken to explain the computation of WRF. The displayed sentence consists of entities belonging to both failure location and type. The calculations of P, T, P_c, T_c, M_c, R1-Recall, R1-Precision, and R1-F1 must be performed, as described in Table 3 for both classes. But in WRF computation, if there is more than one class in the given example, one more class needs to be considered. The calculation for the *combined* class is shown in Table 4. During the formation of strings P and T for the *combined* class, entities belonging to all the classes are considered, unlike the computation involved for individual classes. The *combined* class considers misclassifications related to entities to be correct, except when they are misclassified as *other*. The individual classes (failure location and failure type) consider misclassifications related to entities to be equivalent to misclassifications classified as *other*.

R1-F1 for entity recognition is also affected by repetitive entities (Table 3).

When compared to Table 3, the additional step in Table 4 is to eliminate the repetitive unigrams after forming the strings P and T, which results in P_u and T_u respectively. R1-F1 is computed for all the individual classes and the *combined* class separately by using the R1-Recall and R1-Precision formulas described in Table 3. The WRF is computed by taking a weighted sum of all the R1-F1 values. The example in Table 4 has two classes (ignoring IOB2 format). By including the *combined* class, the total number of classes involved for WRF calculation is three. γ_1, γ_2, and γ_3 are used as three weights for the weighted summing step of WRF. The correct weight-based parameter configuration require domain-specific expert knowledge depending on the underlying use case. The weight of each class determines the importance of that class. If the identification entities is more crucial than the correct classification, the weight of the *combined* class is defined to outweight the weight of individual classes. If the correct classification of entities is more important than just identification, then the weight of the *combined* class is lower than individual class weights. We used two sets of weights in our analysis: WRF_{strict} and $WRF_{lenient}$. WRF_{strict} assigns an equal weight of 0.33 to all three classes, while $WRF_{lenient}$ gives γ_1 and γ_2 a weight of 0.25 each and assigns double weightage (0.5) to γ_3.

Subsection 3.2 shows the problem by computing the F1 score. The evaluation metric of both sentences should be the same since their predictions convey the same information (Table 2). Due to the repetitive and redundant entity predictions, the F1 score gets influenced, resulting in different values. Table 5 describes how the issue is solved by using WRF. Table 5 first outlines the procedure of creating string P by concatenating the predicted failure location entities, and string T by concatenating the

[1]Hereafter, all mentions of WRF represent the weighted R1-F1.

E.g. Subsection 3.1 (Multiclass example)	Weighted R1-F1 score for entity recognition		
	Failure Location	**Failure Type**	**Failure Location & Type** (*combined*)
Form string P from predicted entities	ekmv scroll scroll tip	faulty partially melted too high temperatures	faulty ekmv scroll scroll tip partially melted too high temperatures
Form string T from target entities	scroll tip	partially melted too high temperatures	scroll tip partially melted too high temperatures
P_u: Keep only unique words in P	ekmv ~~scroll~~ scroll tip	faulty partially melted too high temperatures	faulty ekmv ~~scroll~~ scroll tip partially melted too high temperatures
T_u: Keep only unique words in T	scroll tip	partially melted too high temperatures	scroll tip partially melted too high temperatures
R1-F1 with P_u & T_u (Table 3)	0.80	0.91	0.88
Interpretation	Treats misclassification of entities as equal to misclassifications belonging to „other“ class.		Treats misclassification of entities as correct except misclassifications as „other“ class.
Weighted Rouge-1 F1-Score (WRF)	$\gamma_1 *$ R1-F1$_{\text{Failure Location}}$ + $\gamma_2 *$ R1-F1$_{\text{Failure Type}}$ + $\gamma_3 *$ R1-F1$_{\text{Combined}}$ $\quad (C = 2)$ with $\sum_{i=1}^{K} \gamma_i = 1$, where $K = \begin{cases} C & \text{, if } C = 1 \\ C+1 & \text{, if } C > 1 \end{cases}$ and C is the number of classes (ignoring B- and I- prefixes)		
Motivation of $\gamma_1, \gamma_2, \gamma_3$	$\gamma_1, \gamma_2, \gamma_3$ are weighting factors which require domain-specific expert knowledge: $\gamma_3 > 0.333$, if the identification of entities is more important than the correct classification. $\gamma_3 <= 0.333$, if the correct classification is more important than just detection of entities.		
WRF$_{\text{strict}}$ $\gamma_1 = \gamma_2 = \gamma_3 = 0.333$	$\text{WRF}_{\text{strict}} = \gamma_1 * 0.80 + \gamma_2 * 0.91 + \gamma_3 * 0.88 = 0.86$		if $C > 1$, then $\gamma_{C+1} = \gamma_i$ where $i \in [1, C]$
WRF$_{\text{lenient}}$ $\gamma_1 = \gamma_2 = 0.25, \gamma_3 = 0.50$	$\text{WRF}_{\text{lenient}} = \gamma_1 * 0.80 + \gamma_2 * 0.91 + \gamma_3 * 0.88 = 0.87$		if $C > 1$, then $\gamma_{C+1} = 2 * \gamma_i$ where $i \in [1, C]$

Table 4: WRF-calculation. For presentation reasons, we displayed the calculation just in a simplified version of failure location and failure type, instead of using the IOB2 format. Nevertheless, the WRF calculation can be done with every entity recognition annotation format. $\gamma_1, \gamma_2, \gamma_3$ weighting factors can be chosen depending on the application domain by an expert. WRF can ensure prioritization of the identification of entities over the classification correctness of entities depending on the needs of the use case.

Prediction 1: Repair costs (parts and labour) are often very high, since the workshop does not know which is the faulty location and then also replaces the ekmv **FP** or replaces because of the consequential damage to the scroll **FP**, (scroll tip **TP** is partially melted) by too high temperatures.		**Prediction 2**: Repair costs (parts and labour) are often very high, since the workshop does not know which is the faulty location and then also replaces the ekmv **FP** or replaces because of the consequential damage to the *scroll*, (scroll tip **TP** is partially melted) by too high temperatures.
String **T**	scroll tip	scroll tip
String **P**	ekmv scroll scroll tip	ekmv scroll tip
P_u: Keep only unique words in P	ekmv ~~scroll~~ scroll tip	ekmv scroll tip
T_u: Keep only unique words in T	scroll tip	scroll tip
WRF $(C = 1)$ **and** $\gamma_1 = 1.0$	**WRF = 0.80**	**WRF = 0.80**
Insight	The classification result of the repetitive scroll occurrence is not affecting the WRF.	

Table 5: How WRF solves the issue.

-	F1-score	Weighted Rouge-1 F1-Score (WRF_{strict})
Failure Location (FL)	0.777	0.866
Failure Type (FT)	0.821	0.842
Failure Location and Type (Combined) (FC)	-	0.872
$Mean_{FL,FT,FC}$	(0.795)	0.860

Table 6: Experimental results based on the test set described in 4.1. We report the metrics F1-score and $WRF_{unigram}$ scores. The regularization terms γ_1, γ_2, and γ_3 are set to 0.333 (equally weighted). The score in bracket is calculated without regularization term γ and without consideration of $F_{Combined}$.

target failure location entities. The strings P_u and T_u were generated by removing any duplicate unigrams. Computing WRF as demonstrated in Table 4 involves calculating the weighted sum of R1-F1 for all classes, including the *combined* class. There is no distinction between WRF_{strict} or $WRF_{lenient}$ in Table 5 since there is only one class involved (failure location). The insight obtained from Table 5 is, WRF is not affected by repetitive and redundant entities since the resulting WRF metric for both prediction examples is equal.

4 Experimentation

4.1 Training

We used 4 NVIDIA Tesla V100 PCIE 16GB GPUs for the fine-tuning of the *BERT base-uncased* model to the respective AER downstream task over 12 epochs with patience of 4 for early-stopping. The batch size for training was set to 16 with a maximum input sequence of 512. The labeled dataset consists of 5,487 sentences. We defined a 4,005 training, 475 validation, and 1,007 test set split. AdamW was chosen as an optimizer with a learning rate of 1e-4. The learning rate is decreased by a factor of 0.1 whenever the loss decrease stops.

4.2 Quantitative Evaluation

The experiments are performed with the AER test data set by using the fine-tuned BERT-base uncased model. We report the metrics F1-score and WRF. The results are shown in Table 6.

4.3 Qualitative Evaluation

In order to validate the WRF evaluation score, we will randomly select a subset of 60 samples from the test set and use the supervised model according to subsection 4.1 to predict the entities from this subset. We will then compare the predictions of the model using the WRF and F1 metrics. According to subsection 3.3, WRF is expected to evaluate the model predictions more accurately than F1, because F1 can be impacted by the existence of redundant and repeated entities. Three major cases for evaluation comparison are displayed in Tables 10 - 14. The F1 score is higher than the WRF score in 7 out of 60 (11,67%) cases. If the model's predictions of repeating entities are also correctly classified, i. e., the target labels also contain repetitive entities, then F1_micro overestimates the model's performance, leading to a larger value (Table 13).

A higher WRF score compared to the F1 score was identified in 25 out of 60 sentences (41,67%). The model does not predict repetitive entities in a correct way. The calculation of WRF does not take mispredicted redundant entities into account. Furthermore, the F1 score declares a mispredicted entity within a correctly labeled sequence of entities as an overall failure of the entire sequence (Table 10 - Table 12). The WRF and f1 score are equal if both, the prediction entity set and target entity set matches (Table 14). We identified 28 out of the 60 examples (46,66%) for this use case. Additional examples cannot be provided due to confidentiality constraints.

5 Conclusion

We present to the research community a new metric called WRF to fill the evaluation gap in the entity recognition evaluation. We used a weighted form of the Rouge unigram F1, which differentiates between misclassification and non-recognition of entities. WRF is also able to handle redundant entities. The newly developed metric was applied successfully within AER. It is beneficial for the practical use case to make it more focused on correct classification or just the identification of entities. It is possible to optimize the weights of WRF according to its practical use case by the parameters $\gamma_{1,2,3}$.

References

ACE08. 2008. Automatic content extraction 2008 evaluation plan (ace 08) assessment of detection and recognition of entities and relations within and across documents.

Mohamed Ben Jannet, Martine Adda-Decker, Olivier Galibert, Juliette Kahn, and Sophie Rosset. 2014. ETER : a new metric for the evaluation of hierarchical named entity recognition. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Nancy Chinchor and Beth Sundheim. 1993. MUC-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

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.

Jinlan Fu, Pengfei Liu, and Graham Neubig. 2020. Interpretable multi-dataset evaluation for named entity recognition. *CoRR*, abs/2011.06854.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

David M. W. Powers. 2015. What the f-measure doesn't measure: Features, flaws, fallacies and fixes. *CoRR*, abs/1503.06410.

Natalie Schluter. 2017. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 2, page 41–45, United States. Association for Computational Linguistics. The 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 ; Conference date: 03-04-2017 Through 07-04-2017.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Abhishek Sharma, Amrita, Sudeshna Chakraborty, and Shivam Kumar. 2022. Named entity recognition in natural language processing: A systematic review. In *Proceedings of Second Doctoral Symposium on Computational Intelligence*, pages 817–828, Singapore. Springer Singapore.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

A Appendix

Usecase	Text string	Target Entity string	Prediction Entity string
1	...	...	...
	scroll	B-Failure_Loc	B-Failure_Loc
	tip	I-Failure_Loc	I-Failure_Loc
	is	O	O
	partially	B-Failure_Type	B-Failure_Type
	melted	I-Failure_Type	I-Failure_Type
	by	O	O
	too	B-Failure_Type	B-Failure_Type
	high	I-Failure_Type	I-Failure_Type
	temperatures	I-Failure_Type	I-Failure_Type
2	...	...	...
	scroll	B-Failure_Loc	B-Failure_Loc
	tip	I-Failure_Loc	I-Failure_Loc
	is	O	O
	partially	B-Failure_Type	B-Failure_Type
	melted	I-Failure_Type	I-Failure_Type
	by	**O**	**B-Failure_Loc**
	too	B-Failure_Type	B-Failure_Type
	high	I-Failure_Type	I-Failure_Type
	temperatures	I-Failure_Type	I-Failure_Type
3	...	...	...
	scroll	**B-Failure_Loc**	**O**
	tip	**I-Failure_Loc**	**O**
	is	O	O
	partially	B-Failure_Type	B-Failure_Type
	melted	I-Failure_Type	I-Failure_Type
	by	O	O
	too	B-Failure_Type	B-Failure_Type
	high	I-Failure_Type	I-Failure_Type
	temperatures	I-Failure_Type	I-Failure_Type

Table 7: Use cases that can be dealt with the metrics by CoNLL2003. The first use-case describes the full match of the target string and the prediction string. The second use-case describes the hypothecation of an entity, while the third use-case deals with the case of a missing entity prediction. Only a segment of the complete W&G sentence (Section 3.1) is listed in tabular form.

Usecase	Text string	Target Entity string	Prediction Entity string
	...	...	...
	scroll	**B-Failure_Loc**	**B-Failure_Type**
	tip	**I-Failure_Loc**	**I-Failure_Type**
	is	O	O
4	partially	B-Failure_Type	B-Failure_Type
	melted	I-Failure_Type	I-Failure_Type
	by	O	O
	too	B-Failure_Type	B-Failure_Type
	high	I-Failure_Type	I-Failure_Type
	temperatures	I-Failure_Type	I-Failure_Type
	...	...	...
	scroll	B-Failure_Loc	B-Failure_Loc
	tip	I-Failure_Loc	I-Failure_Loc
	is	**O**	**B-Failure_Type**
	partially	**B-Failure_Type**	**I-Failure_Type**
5	melted	**I-Failure_Type**	**I-Failure_Type**
	by	O	O
	too	B-Failure_Type	B-Failure_Type
	high	I-Failure_Type	I-Failure_Type
	temperatures	I-Failure_Type	I-Failure_Type
	...	...	...
	scroll	B-Failure_Loc	B-Failure_Loc
	tip	I-Failure_Loc	I-Failure_Loc
	is	**O**	**B-Failure_Loc**
	partially	**B-Failure_Type**	**I-Failure_Loc**
6	melted	**I-Failure_Type**	**I-Failure_Loc**
	by	O	O
	too	B-Failure_Type	B-Failure_Type
	high	I-Failure_Type	I-Failure_Type
	temperatures	I-Failure_Type	I-Failure_Type

Table 8: Use cases that can be dealt with the metrics by SemEval'13. The fourth use-case describes the wrong assignment of a predicted entity type. The fifth use-case describes the wrong definition of entity boundaries, while the sixth use-case deals with both a wrong entity type assignment and a wrong boundary definition. Only a segment of the complete W&G sentence (Section 3.1) is listed in tabular form.

Usecase	Text string	Target Entity string	Prediction Entity string
7	...	...	...
	scroll	**O**	**B-Failure_Loc**
	...	...	...
	scroll	**B-Failure_Loc**	**B-Failure_Loc**
	tip	**I-Failure_Loc**	**I-Failure_Loc**
	is	O	O
	partially	B-Failure_Type	B-Failure_Type
	melted	I-Failure_Type	I-Failure_Type
	by	O	O
	too	B-Failure_Type	B-Failure_Type
	high	I-Failure_Type	I-Failure_Type
	temperatures	I-Failure_Type	I-Failure_Type

Table 9: Use case which can not be dealt with CoNLL2003 or SemEval'13 metrics. Only a segment of the complete W&G sentence (Section 3.1) is listed in tabular form.

Prediction	steering wheel trim on left side trim not flush - sticking outward (looks warped) removed and replaced drivers steering wheel - ok . cv
Target	steering wheel trim on left side trim not flush - sticking outward (looks warped) removed and replaced drivers steering wheel - ok . cv
Calculated F1-Score	**0.440**
Calculated WRF with $\gamma_{1,2,3} = 0.333$	**0.820**

Table 10: Case 1.1: WRF > F1-Score. If the model's predictions for repeating entities are incorrectly classified, i. e., the target labels do not contain repetitive entities, then $F1_{micro}$ underestimates the model's performance and produces a lower value. The second occurrence of the steering wheel is wrongly predicted as an entity by the model, unlike the first occurrence. This sentence has been artificially generated to simulate typical customer feedback patterns.

Prediction	overhead control panel will not close properly ; replaced overhead control pane for sunglasses compartment compartment would not clos e completely.
Target	overhead control panel will not close properly ; replaced overhead control pane for sunglasses compartment compartment would not clos e completely.
Calculated F1-Score	**0.530**
Calculated WRF with $\gamma_{1,2,3} = 0.333$	**0.830**

Table 11: Case 1.2: WRF > F1-Score. The entity will not close properly predicted by the model will be misclassified since the $F1_{micro}$ score looks for a perfect match of the whole entity and the corresponding target entity is only not close properly. WRF_{strict} will therefore be higher in this situation. This sentence has been artificially generated to simulate typical customer feedback patterns.

Prediction	blower has a noise ; rumbling noise ; blower motor ;r & r glovebox and removed old blower motor due to it being noisy . replace d with a new blower motor and operated toverigy the repair .
Target	blower has a noise ; rumbling noise ; blower motor ;r & r glovebox and removed old blower motor due to it being noisy . replace d with a new blower motor and operated toverigy the repair .
Calculated F1-Score	**0.910**
Calculated WRF with $\gamma_{1,2,3} = 0.333$	**1.000**

Table 12: Case 1.3: WRF > F1-Score. If a model incorrectly classifies an entity but that entity is part of another entity that was correctly classified, then $F1_{micro}$ underestimates the model's performance. For example, blower is a misclassified entity, but blower motor is a correctly classified entity. Intuitively, the model should not be penalized in this situation, but $F1_{micro}$ underestimates the model's performance. This sentence has been artificially generated to simulate typical customer feedback patterns.

Prediction	guest states rumble coming out of the fan system at a higher level ofspeed , like a chattering ; found blower motor imbalance , replace blower motor.
Target	guest states rumble coming out of the fan system at a higher level ofspeed , like a chattering ; found blower motor imbalance , replace blower motor.
Calculated F1-Score	**0.910**
Calculated WRF with $\gamma_{1,2,3} = 0.333$	**0.840**

Table 13: Case 2: WRF < F1-Score. The WRF calculation leads to a lower metric value compared to the F1-Score. If the model's predictions of repeating entities are also correctly classified, i. e., the target labels also contain repetitive entities, then $F1_{micro}$ overestimates the model's performance, leading to a larger value. For example, blower motor is the repeated entity predicted by the model, and all occurrences are correctly classified in both cases. This sentence has been artificially generated to simulate typical customer feedback patterns.

Prediction	left front seat cushion cover cracking ; verified leather is starting to crack ; replaced seat bottom leather on drivers seat.
Target	left front seat cushion cover cracking ; verified leather is starting to crack ; replaced seat bottom leather on drivers seat.
Calculated F1-Score	**1.000**
Calculated WRF with $\gamma_{1,2,3} = 0.333$	**1.000**

Table 14: Case 3: WRF = F1-Score. The prediction entity string matches the target entity string. Both, WRF and F1 score calculate the maximum result. This sentence has been artificially generated to simulate typical customer feedback patterns.