Evaluating Sequence Labeling on the basis of Information Theory

Enrique Amigó, Elena Álvarez-Mellado,
Julio Gonzalo and **Jorge Carrillo-de-Albornoz**
UNED NLP & IR Research Group - `nlp.uned.es`
UNED School of Computer Science, Madrid, Spain
`{enrique,elena.alvarez,julio,jcalbornoz}@lsi.uned.es`

Abstract

Various metrics exist for evaluating sequence labeling problems (strict span matching, token oriented metrics, token concurrence in sequences, etc.), each of them focusing on certain aspects of the task. In this paper, we define a comprehensive set of formal properties that captures the strengths and weaknesses of the existing metric families and prove that none of them is able to satisfy all properties simultaneously. We argue that it is necessary to measure how much information (correct or noisy) each token in the sequence contributes depending on different aspects such as sequence length, number of tokens annotated by the system, token specificity, etc. On this basis, we introduce the **S**equence **L**abelling **I**nformation **C**ontrast **M**odel (SL-ICM), a novel metric based on information theory for evaluating sequence labeling tasks. Our formal analysis and experimentation show that the proposed metric satisfies all properties simultaneously.

1 Introduction

Span identification tasks are a type of NLP problem in which relevant spans of text are retrieved from sentences (Papay et al., 2020). Named entity recognition (NER) is a prime example of a span identification task, but there are many others in the field of Information Extraction (such as multiword expression extraction, time expression extraction, term extraction, toxic language detection, opinion mining, etc.) and in language processing (semantic role labeling, chunking, etc.). Span identification tasks are usually framed as a sequence labeling problem, in which each token of the sequence receives a tag and the tag assignment is done contextually: each assigned tag will depend on the nature of the surrounding tokens and tags, and token-level prefixes (such as BIO or BILOU) will be added to the tag to denote the boundary of the span (Ramshaw and Marcus, 1999; Ratinov and Roth, 2009).

Multiple evaluation approaches have been proposed for sequence labeling, such as strict span matching, token-oriented metrics, token concurrence in sequences, etc. Each of these approaches addresses the problem in a different way, and consequently they reward and penalize outputs differently. However, the mismatch between the nature of the problem (in which *spans of text* are the unit of interest) and its token-based implementation leads to inescapable problems when evaluating span-identification tasks: strict span-based evaluation will disregard altogether any predicted spans that partially overlap with the gold standard. On the other hand, token-based evaluation may not adequately account for span boundaries and it displays an imperfect treatment of overlapping spans. In addition, none of the existing metrics takes into account the degree of informativeness of each token.

We argue that the key issue when evaluating span identification tasks is to measure how much information (correct or noisy) each token contributes to the span, depending on different aspects such as span length, the number of tokens annotated by the system, the token specificity, etc. On this basis, we first define a set of formal properties and we characterize existing evaluation metrics according to them (section 2). Our analysis shows that existing metrics sacrifice some properties in favor of others, and that some properties are not captured by any of the existing metrics.

We claim that the natural way of addressing this problem is in terms of information quantities. We introduce SL-ICM (**S**equence **L**abelling **I**nformation **C**ontrast **M**odel), an information theory based metric for evaluating sequence labeling problems (section 3). This metric is grounded on the ICM similarity measure, which is a linear combination of the single and joint information content of the elements in the comparison. Finally, we conduct a series of experiments on manually selected

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 27849–27860
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

examples and on real existing data in order to systematically assess how metrics behave on different scenarios (Section 4). Our results prove that SL-ICM is the only metric capable of satisfying all properties simultaneously[1].

2 A Formal Analysis of Sequence Labeling Metrics

In this section we introduce a set of formal properties for sequence labeling evaluation and analyze whether existing metrics satisfy them.

The following notation will be used to formalize the metrics: let W be the ordered set of tokens that make up a text, and let s be a subsequence (not necessarily continuous) of W. We denote $\hat{S}_t$ as the set of sequences annotated by a system with label t, and S_t as the annotated set in the gold standard. A sequence labeling evaluation metric measures the similarity between the sets of sequences S_t and $\hat{S}_t$ for all labels $t \in T$.

In order to formalize the properties, we will denote a gold standard sequence annotation as $\{(x_1, y_1, t_1), ..., (x_n, y_n, t_n)\}$ where x_i, y_i represent the start and end token positions of the i^{th} labeled sequence and t_i represents the assigned label. Then, a system output represented as $\{(x_3, y_3 - 1, t)\}$ is an output that includes the third sequence annotated as t but without the last token of the reference sequence (position y_3). For the sake of simplicity, we formalize the properties on continuous sequences. We use the symbol ">" to express that one sequence of labels should receive a higher score than another. Table 1 summarizes the properties satisfied by each metric, and Table 2 shows particular examples for each property.

2.1 Strict Span-based Evaluation

Exact matching based evaluation can be formalized with the following two properties, which state that the score should be higher when more correct sequences are predicted, and lower when more incorrect sequences are predicted.

Property 1. [CORRECT SEQUENCE MONOTONICITY] *Adding a correct sequence to the system output should increase its score:*

$$\{(x_1, y_1), ..., (x_{j-1}, y_{j-1}), (x_j, y_j)\} > \{(x_1, y_1), ..., (x_{j-1}, y_{j-1})\}$$

Property 2. [WRONG SEQUENCE MONOTONICITY] *Adding an incorrect sequence to the output should decrease the system score. Assuming $y_i < k < l < x_{i+1}$:*

$$\{...,(x_i, y_i), (k, l), (x_j, y_j)...\} < \{..., (x_i, y_i), (x_j, y_j)...\}$$

F-measure over exact matching spans (Tjong Kim Sang and De Meulder, 2003) complies with these two properties. It is a combination of the following precision and recall metrics:

$$\text{Span Pre.}(\hat{S}_t, S_t) = \frac{\sum_t |\hat{S}_t \bigcap S_t|}{\sum_t |\hat{S}_t|},$$
$$\text{Span Rec.}(\hat{S}_t, S_t) = \frac{\sum_t |\hat{S}_t \bigcap S_t|}{\sum_t |S_t|}$$

Awasthy et al. (2020) proposed another exact matching based metric for NER tasks called *Span Similarity*. However, it can be proved that it is equivalent to span F-measure.

$$\text{SpanSim}(S_t, \hat{S}_t) = \frac{2 \cdot |S_t \bigcap \hat{S}_t|}{2 \cdot |S_t \bigcap \hat{S}_t| + |S_t \setminus \hat{S}_t| + |\hat{S}_t \setminus S_t|}$$

These metrics, and in general the rest of metrics described in this paper, can be computed for each label t and then averaged (macro average) or considering all annotated sequences as a whole. The first approach avoids underestimating infrequent labels.

2.2 Token-based Evaluation

The main drawback of exact matching metrics is that they do not capture partial sequence predictions, which is especially relevant when dealing with long sequences. The following properties formalize the ability to capture single tokens within sequences. Let ϵ be an integer value such that $0 \leq \epsilon \leq y_i - x_i$.

Property 3. [OVERLAP MONOTONICITY] *Adding a correct token to a sequence prediction should increase the system score. Assuming $\epsilon' > \epsilon$:*

$$\{..., (x_i + \epsilon, y_i \pm \epsilon), ...\} > \{..., (x_i + \epsilon', y_i \pm \epsilon), ...\}$$

Property 4. [NOISE MONOTONICITY] *Adding an incorrect token to a sequence prediction should decrease the score. Assuming $\epsilon' > \epsilon$:*

$$\{..., (x_i - \epsilon, y_i \pm \epsilon), ...\} > \{..., (x_i - \epsilon', y_i \pm \epsilon), ...\}$$

[1] The metric software and related documentation are available at: https://sites.google.com/view/enriqueamigo

Table 1: Metrics and Formal Properties

Metrics	Formal Properties										
	Correct Seq. Mon.	Wrong Seq. Mon.	Overlap Mon.	Noise Mon.	Seq. Homogeneity	Seq. Completedness	Seq. Len. vs. Capt. Words	Seq. Len. vs. Noisy Words	Overlap Increasing Mon.	Noise Increasing Mon.	Token Informativeness.
Span Matching based Metrics											
F-measure over exact matches	✔	✔	✗	✗	✔	✔	✗	✗	✗	✗	✗
Token based metrics											
Token Tag F-measure (IO)	✔	✔	✔	✔	✗	✗	✗	✗	✗	✗	✗
Token Tag F-measure (BIO,BIOE)	✔	✔	✗	✔	✔	✔	✗	✗	✗	✗	✗
Token Tag F-measure (IO*)	✔	✔	✔	✔	✔	✔	✗	✗	✗	✗	✗
Link based metrics											
Link based F-measure	✔	✔	✔	✔	✔	✔	✗	✗	✗	✗	✗
BCubed metrics (extended)	✔	✔	✔	✔	✔	✔	✗	✗	✗	✔	✗
Partial Matching based Metrics											
Intersection-based F-measure	✔	✔	✔	✔	✗	✗	✔	✔	✗	✔	✗
Information-based Metrics											
Information Contrast Model (ICM)	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

Tag accuracy and token based F1-measure satisfy these two properties. Under these metrics, sequence labeling is evaluated as a token classification task with one category per tag plus *"no tag"* (Ratnaparkhi, 1996; Molina et al., 2016).

$$\text{Token Pre.}(\hat{S}_t, S_t) = \frac{\sum_{\substack{\hat{s}\in\hat{S}_t \\ s\in S_t}} |\hat{s} \cap s|}{\sum_{\hat{s}\in\hat{S}_t} |\hat{s}|}$$

$$\text{Token Rec.}(\hat{S}_t, S_t) = \frac{\sum_{\substack{\hat{s}\in\hat{S}_t \\ s\in S_t}} |\hat{s} \cap s|}{\sum_{s\in S_t} |s|}$$

However, considering sequence labeling as a token classification task has the limitation of missing boundaries in contiguous sequences. The following properties focus on this issue:

Property 5. [Sequence Homogeneity] *Correctly splitting a predicted sequence into two contiguous sequences which are in the gold standard must increase the score. Supposing* $x_i < y_i < x_j < y_j$ *and* $x_j = y_i + 1$*:*

$$\{..., (x_i, y_i), (x_j, y_j)...\} > \{..., (x_i, y_j)...\}$$

Property 6. [Sequence Completeness] *Splitting a correct sequence prediction into two contiguous sequences must decrease the score. Supposing* $x_i < k < y_i$*:*

$$\{..., (x_i, y_i)...\} > \{..., (x_i, k), (k + 1, y_i)...\}$$

In token-based prediction, this problem has commonly been solved by using prefixed labels to indicate whether the token is at the beginning or inside the sequence (BIO encoding), and additionally including prefixes for tokens at the end of sequences (BIOE encoding) or for single token spans (BILOU, BIOES or BMES encoding). However, this may result in certain cases in Overlap Monotonicity not being satisfied: a system using BIO encoding that identifies only a single token in the middle of a sequence may not be rewarded, since the identified token will be labeled with a B tag instead of an I tag.

In order to capture sequence boundaries while still allowing for partial matching, Esuli and Sebastiani (2010) proposed an alternative format that considers whitespaces as taggable units that can be in or out of the sequences. We refer to this as IO* format in Table 1.

2.3 Link-based metrics

Another way of capturing sequence boundaries is treating the labeled sequences as links between tokens (Vilain et al., 1995; Schneider et al., 2014). While in the case of exact matching-based metrics the sequence labeling problem is understood as a classification problem, link-based metrics frame sequence labeling as a clustering problem. Two tokens are linked if there is an annotated sequence labeled as t in which both tokens appear. This allows for partial matches being taken into account.

$$\begin{aligned} w_i \hat{\text{—}} w_j &\equiv \exists t \in T, s \in \hat{S}_t(w_i, w_j \in s) \\ w_i \text{—} w_j &\equiv \exists t \in T, s \in S_t(w_i, w_j \in s) \end{aligned}$$

$$\text{LinkPrecision}(S_t, \hat{S}_t) = \frac{|\{i, j : w_i \hat{\text{—}} w_j \wedge w_i \text{—} w_j\}|}{|\{i, j : w_i \hat{\text{—}} w_j\}|}$$

$$\text{LinkRecall}(S_t, \hat{S}_t) = \frac{|\{i, j : w_i \hat{\text{—}} w_j \wedge w_i \text{—} w_j\}|}{|\{i, j : w_i \text{—} w_j\}|}$$

The main problem with link F-measure is the combinatorial explosion of links in large sequences, which may bias the evaluation.

BCubed metrics partially solve this problem by calculating link recall and precision at the token level (Amigó et al., 2009). BCubed precision measures the quality of the links established by the system for each token, and BCubed recall represents the proportion of links covered for that token compared to those present in the gold standard. Letting W_t and $\hat{W}_t$ be the sets of tokens annotated in any sequence in the system output and the gold standard respectively:

$$\text{BC-Pre.}(\hat{S}_t, S_t) = \frac{1}{|W_{S_t}|} \sum_{w_i \in \hat{W}_t} \frac{|\{j : w_i \hat{\text{—}} w_j \wedge w_i \text{—} w_j\}|}{|\{j : w_i \hat{\text{—}} w_j\}|}$$

$$\text{BC-Rec.}(\hat{S}_t, S_t) = \frac{1}{|W_{S_t}|} \sum_{w_i \in W_t} \frac{|\{j : w_i \hat{\text{—}} w_j \wedge w_i \text{—} w_j\}|}{|\{j : w_i \text{—} w_j\}|}$$

2.4 Partial Matching based Metrics

Link-based metrics have certain limitations. Since the evaluation is performed at token-pair level, the sequence length is ignored. But the shorter a sequence is, the more a single token provides information about the sequence: capturing the last token *mind* in the span *Eternal sunshine of the spotless mind* contributes less information than capturing the last token *York* in the span *New York*. Therefore, predicting a correct token (or an incorrect one) should have less impact on a long sequence than on a short sequence. The following two properties formalize this idea. Assuming the i^{th} sequence in the gold standard be longer than the j^{th} sequence, i.e. $y_i - x_i > y_j - x_j$:

Property 7. [SEQUENCE LENGTH VS. CAPTURED WORDS] *The benefit of predicting a correct term is higher in shorter sequences.*

$$\begin{aligned} &\{..., (x_i + 1, y_i), ..., (x_j, y_j), ...\} \\ &\quad > \{..., (x_i, y_i), ..., (x_j + 1, y_j), ...\} \end{aligned}$$

Property 8. [SEQUENCE LENGTH VS. NOISY WORDS] *The penalty of predicting an incorrect term is higher in shorter sequences.*

$$\begin{aligned} &\{...,(x_i - 1, y_i), ..., (x_j, y_j), ...\} \\ &\quad > \{..., (x_i, y_i), ..., (x_j - 1, y_i), ...\} \end{aligned}$$

Link F-measure does not consider sequence length and cannot satisfy these properties. BCubed metrics do, but produce the opposite effect to that described in these properties[2].

In order to correctly account for sequence length, it is necessary to evaluate at sequence level instead of token or link level, while keeping the sensitivity to partial sequence prediction. Metrics from MUC evaluation campaigns distinguished between total or partial hits, but without considering the length of the sequence (Chinchor and Sundheim, 1993).

Johansson and Moschitti (2013) proposed an intersection-based precision and recall to evaluate the detection of polarity expressions. These metrics have subsequently been used for propaganda detection (Da San Martino et al., 2019) and toxic span detection (Pavlopoulos et al., 2021). First, they formalize the token coverage on gold standard sequences, i.e., span coverage $\left(C(\hat{s}, s, h) = \frac{|\hat{s} \cap s|}{h}\right)$ where h is normalization factor which represents the predicted or true sequence length. Then the intersection-based precision and recall are defined as the span coverage C regarding the sequence length sum in the system output and the gold standard respectively:

$$\text{Int-Based-Pre}(\hat{S}_t, S_t) = \frac{1}{|\hat{S}_t|} \sum_{\substack{\hat{s} \in \hat{S}_t \\ s \in S_t}} C(\hat{s}, s, |\hat{s}|)$$

$$\text{Int-Based-Rec}(\hat{S}_t, S_t) = \frac{1}{|S_t|} \sum_{\substack{\hat{s} \in \hat{S}_t \\ s \in S_t}} C(\hat{s}, s, |s|)$$

However, since these metrics aggregate the intersections of each sequence of the system output with different sequences of the gold standard and vice versa, SEQUENCE HOMOGENEITY and SEQUENCE COMPLETENESS are not satisfied. Another partial matching based metric is the Loss per Sequence Nguyen and Guo (2007), but it exclusively focuses on recall.

[2] For instance, missing a token in a sequence of length 10 reduces the recall associated with that token to zero and the recall associated with the rest of the tokens in the sequence by $1/10$ (i.e. $1/10 \cdot 9 = 9/10$). This effect is greater than in 3-long sequences, in which case it would affect $1/3$ recall on two tokens (i.e. $1/3 \cdot 2 = 2/3$).

Table 2: Formal Property Examples: Prediction A is better than prediction B for each of the properties.

Property	Gold	Sys. A	Sys. B
Correct Seq. Mon.	1-3 10-13 20-23	1-3 10-13	1-3
Wrong Seq. Mon.	1-3 10-13 20-23	1-3 10-13	1-3 10-13 30-33
Overlap Mon.	1-7 10-13 20-23	1-6 10-13 20-23	1-5 10-13 20-23
Noise Mon.	1-7 10-13 20-23	1-8 10-13 20-23	1-9 10-13 20-23
Seq. Homogeneity	1-5 6-7 10-13 20-23	1-5 6-7 10-13	1-7 10-13
Seq. Completedness	1-7 10-13 20-23	1-7 10-13	1-5 6-7 10-13
Seq. L. vs. Capt. W.	10-16 20-23	10-15 20-23	10-16 20-22
Seq. L. vs. Noisy W.	10-16 20-23	10-17 20-23	10-16 20-24
Overlap Inc. Mon.	10-16 20-26	10-14 20-25	10-13 20-26
Noise Inc. Mon.	10-16 20-26	10-17 20-31	10-18 20-30

3 SL-ICM: An Information Theory based Metric

Despite all properties defined in the previous section, there are still some aspects which no metric captures. One is that the credit earned by a system for detecting the n^{th} token in a sequence should depend on n. For instance, if the ground truth is *"Pope John Paul II"*, then adding *"Paul"* to *"Pope John"* is more important than adding *"II"* to *"Pope John Paul"*, because *"Paul"* adds one third of the content while *"II"* adds one fourth.

This idea is formalized by the following properties. Let the i^{th} and the j^{th} sequences in the gold standard be of the same size ($y_i - x_i = y_j - x_j$) and let n be an integer:

Property 9. [Overlap Increasing Monotonicity] *The benefit of capturing a token decreases with the amount of tokens already captured in the sequence:*

$$\{..., (x_i, y_i - n), ..., (x_j, y_j - n), ...\} \\ > \{..., (x_i, y_i - n - 1), ..., (x_j, y_j - n + 1), ...\}$$

In the same way, each token incorrectly predicted is increasingly less harmful:

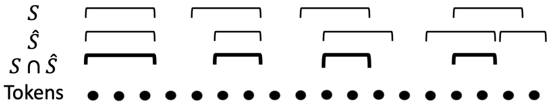

Figure 1: Illustration of sequence set intersection.

Property 10. [Noise Increasing Monotonicity] *The penalty for incorrect term predictions decreases with the number of noisy words already predicted in the sequence:*

$$\{..., (x_i, y_i + n), ..., (x_j, y_j + n + 2), ...\} \\ > \{..., (x_i, y_i + n + 1), ..., (x_j, y_j + n + 1), ...\}$$

BC-metrics and intersection-based F-measure satisfy Noise Increasing Monotonicity because precision decreases more smoothly as we introduce noisy tokens into the predicted sequence, but they do not satisfy Overlap Increasing Monotonicity.

Another aspect that previous metrics ignore is token specificity: missing the token *The* in the span *The Ohio State University* is less serious than missing *University*, because the latter is more informative. On the other hand, missing *University* in *Harvard University* is a less serious mistake than missing *Harvard*. In other words, the more specific the token is, the more information it contributes to the sequence. This can be formalized as follows: Being $P(n)$ the probability of the n^{th} token in the language model from which sequences are extracted:

Property 11. [Token Informativeness] *The reward or penalty for noisy tokens decreases with their likelihood. Being $P(y_i)$<$P(y_j)$:*

$$\{..., (x_i, y_i), ..., (x_j, y_j - 1), ...\} \\ > \{..., (x_i, y_i - 1), ..., (x_j, y_j), ...\}$$

At the end of the day, the properties we have just presented revolve around the information quantity that each token contributes (correctly or incorrectly) to the span being predicted. Generally, we can assume that: (i) the longer the sequence, the less information a single token provides; (ii) the more tokens the system has identified, the less a new token provides additional information (or noise) and (iii) the more infrequent or specific the token, the more information (or noise) the token will add. Therefore, our claim is that sequence labeling should be evaluated in terms of textual information quantities.

Table 3: Examples of compliance with formal properties. Each cell contains scores for predictions A/B defined in Table 2. Scores that comply with the properties appear in bold.

Property	Span F	Token-F (IO)	Token-F (BIOE)	Link-F	BC-F	Int-F	SL-ICM
Correct Seq. Mon.	**0.8 / 0.5**	**0.778 / 0.429**	**0.778 / 0.429**	**0.762 / 0.375**	**0.778 / 0.429**	**0.8 / 0.5**	**0.653 / 0.306**
Wrong Seq. Mon.	**0.8 / 0.667**	**0.778 / 0.636**	**0.778 / 0.636**	**0.762 / 0.615**	**0.778 / 0.636**	**0.8 / 0.667**	**0.653 / 0.479**
Overlap Mon.	0.667 / 0.667	**0.966 / 0.929**	**0.897 / 0.857**	**0.921 / 0.843**	**0.934 / 0.871**	**0.976 / 0.95**	**0.979 / 0.954**
Noise Mon.	0.667 / 0.667	**0.968 / 0.938**	**0.903 / 0.875**	**0.923 / 0.85**	**0.938 / 0.883**	**0.979 / 0.962**	**0.991 / 0.983**
Seq. Homogeneity	**0.857 / 0.333**	0.846 / 0.846	**0.846 / 0.692**	**0.848 / 0.737**	**0.846 / 0.737**	0.857 / 0.857	**0.738 / 0.53**
Seq. Completedness	**0.8 / 0.333**	0.846 / 0.846	**0.846 / 0.692**	**0.884 / 0.737**	**0.846 / 0.704**	0.8 / 0.8	**0.692 / 0.535**
Seq. L. vs. Capt. W.	0.5 / 0.5	0.952 / 0.952	0.857 / 0.857	0.899 / 0.944	0.908 / 0.914	**0.963 / 0.933**	**0.969 / 0.947**
Seq. L. vs. Noisy W.	0.5 / 0.5	0.957 / 0.957	0.87 / 0.87	0.905 / 0.938	0.915 / 0.919	**0.968 / 0.947**	**0.987 / 0.979**
Overlap Inc. Mon.	0 / 0.5	0.88 / 0.88	0.72 / 0.8	0.783 / 0.809	0.767 / 0.798	0.88 / 0.88	**0.913 / 0.902**
Noise Inc. Mon.	0 / 0	0.824 / 0.824	0.706 / 0.706	0.659 / 0.671	**0.676 / 0.662**	**0.843 / 0.828**	**0.939 / 0.936**

To account for this, we propose to define a particularization of the general Information Contrast Model (ICM) (Amigó et al., 2020) to the problem of sequence labeling. The derivation of the metric and all related details are described in Appendix A. In this section, we summarize the resulting metric.

In sequence labeling, ICM measures similarity in terms of the amount of information (according to Information Theory) provided by the system output labeled sequence set $\hat{S}$, the gold standard labeled sequence set S, and their intersection:

$$\text{SL-ICM} \equiv 3 \cdot I(\hat{S} \overset{\leftrightarrow}{\cap} S) - I(\hat{S}) - I(S).$$

The intersection $\hat{S} \overset{\leftrightarrow}{\cap} S$ requires mapping the system output sequences in $\hat{S}$ with the corresponding gold standard sequence in S, if they exist. We take the sequence intersections that maximize the information in both directions (as illustrated in Figure 1). The directional intersection $\hat{S} \overset{\rightarrow}{\cap} S$ represents the maximal information intersection between each output sequences in $\hat{S}$ and the true sequences in S:

$$\hat{S} \overset{\leftrightarrow}{\cap} S = (\hat{S} \overset{\rightarrow}{\cap} S) \cap (S \overset{\rightarrow}{\cap} \hat{S})$$
$$\hat{S} \overset{\rightarrow}{\cap} S = \left\{ \hat{s} \cap \operatorname{argmax}_{s \in S} I(\hat{s} \cap s) : \hat{s} \in \hat{S} \right\}$$

SL-ICM can be normalised between 0 and 1 by using the amount of information provided by the ground truth as reference, where 1 corresponds to the maximum score ($I(S)$) and 0 corresponds to an empty output ($-I(S)$):

$$\text{SL-ICM}\, norm(\hat{S}, S) = \frac{\text{SL-ICM}(\hat{S}, S) + I(S)}{2 \cdot I(S)}.$$

On the basis of Information Theory and certain assumptions (see Appendix A), the information content $I(s,t)$ of labeling a sequence $s = (w_1, \ldots, w_l)$ as t can be computed as:

$$I(s,t) = -log \left(\frac{N_t}{N} \prod_{i=1..l} P(w_i)^{\frac{1}{i}} \right)$$

Where N and N_t represent the amount of sequences annotated as t and the total amount of sequences annotated in the gold standard. $P(w_i)$ represents the probability of the token w_i in the corpus. If $P(w_i)$ is unknown, we can assume an unitary information content per token, i.e. $-log(P(w)) = 1$. Therefore:

$$I(s,t) = -log \left(\frac{N_t}{N} \right) + \sum_{i=1..l} \frac{1}{i}$$

According to SL-ICM, a system that does not provide any information (i.e. that makes no predictions) is penalized based on the amount of information provided in the gold standard. A system that generates the same output as the gold standard is rewarded according to the amount of information in the gold standard (i.e. $I(\hat{S}) = 0 \implies \text{SL-ICM} = -I(S)$ and $\hat{S} = S \implies \text{SL-ICM} = I(S)$).

ICM can be normalized between 0 and 1 by using the amount of information provided by the gold standard as reference, where 1 corresponds to the maximum score ($I(S)$) and 0 corresponds to an empty output ($-I(S)$). (See supplementary materials for the metric derivation details and the formal proofs of its satisfied properties.)

4 Empirical results

We now conduct a series of experiments to assess how metrics behave in relation to the properties we

Table 4: Metric scores for gold standard transformations in MULTICONER data set.

Output	Span-F	Token-F (IO)	Token-F (BIOE)	Link-F	BC-F	Int-F	SL-ICM
Gold Standard	1	1	1	1	1	1	1
Sequence split	**0.368**	1	**0.743**	**0.91**	**0.906**	1	**0.704**
Maximum Overlap	0.766	**0.918**	**0.754**	**0.859**	**0.859**	**0.932**	**0.976**
Minimum Overlap	0.766	**0.863**	**0.698**	**0.76**	**0.793**	**0.902**	**0.961**
Long Sequence Truncation	0.961	0.993	0.979	0.984	0.988	**0.996**	**0.998**
Short Sequence Truncation	0.961	0.993	0.979	0.99	0.988	**0.993**	**0.997**
Long Sequence Extension	0.961	0.993	0.98	0.981	0.988	**0.997**	0.999
Short Sequence Extension	0.961	0.994	0.98	0.988	0.989	**0.996**	0.999
-1 Token	0.961	0.993	0.979	0.984	0.988	0.996	0.998
-2 Token	0	-0.007	-0.007	-0.014	-0.01	-0.004	**-0.002**
-3 Token	0	-0.014	-0.014	-0.024	-0.018	-0.008	**-0.005**
+1 Token	0.548	0.926	0.779	0.874	0.866	0.933	0.985
+2 Token	0	**-0.037**	**-0.032**	**-0.072**	**-0.059**	**-0.023**	**-0.006**
+3 Token	0	**-0.072**	**-0.061**	**-0.145**	**-0.107**	**-0.038**	**-0.012**

presented in sections 2 and 3.

4.1 Metric Results on Selected Examples

In our first experiment, we exemplify formal properties compliance with particular cases. Table 2 shows an example of gold standard, system output A and output B that fit the conditions of each formal property definition. The sequences are expressed by their first and last token position. According to the corresponding property (first column), System A should outperform System B.

Table 3 shows the scores achieved by the systems A and B according to each metric in each case. Results that conform to the property (i.e. when System A outperforms System B) are highlighted in bold.

For the sake of simplicity we have excluded TOKEN INFORMATIVENESS since no metric quantifies this aspect except SL-ICM. In the case of SL-ICM, we state a constant token probability $k = P(w) = 1/N$, where N represents the amount of sequences in the gold standard. SL-ICM is normalized with respect to the gold standard information content.

The results displayed in Table 3 are perfectly aligned with the formal analysis carried out above. For instance, all metrics satisfy CORRECT SEQUENCE MONOTONICITY and WRONG SEQUENCE MONOTONICITY; exact span based metrics fail in all other properties, except for SEQUENCE HOMOGENEITY and SEQUENCE COMPLETENESS, etc.

4.2 Metric Results on Real Data

For our second experiment, we used the development set of the Spanish section of the MultiCoNER dataset (Malmasi et al., 2022).

The original set was used as a gold standard. We then performed a series of systematic transformations to the original annotation (adding a noisy token, removing a correct token, shifting the labels one token to the left, etc.) in order to generate synthetic outputs that would allow us to test each of the properties described in sections 2 and 3.

Table 4 shows the metric scores for each of the transformations. For instance, in order to check SEQUENCE COMPLETENESS, we split all multi-token entities into two adjacent sequences. All metrics penalize these mistakes except for token F1 with IO encoding and Int-F, which are not sensitive to sequence boundaries.

In order to assess compliance with OVERLAP MONOTONICITY and NOISE MONOTONICITY, the annotation of sequences of more than 2 tokens was extended one position (Maximum Overlap), so a maximal yet imperfect overlap between the gold standard and the output was achieved. We then applied a Minimum Overlap transformation, where sequences overlap one token only with the gold standard. All metrics are sensitive to varying degrees of overlap, except for span F1. Notice also that token-oriented and link-oriented metrics report a larger difference than Int-F or SL-ICM. The reason is that, for the latter two, the first detected token in a sequence has more effect than the rest, satisfying OVERLAP INCREASING MONOTONICITY and

Noise Increasing Monotonicity).

In order to check Sequence Length vs. Captured Words, we removed the last token of sequences longer than 4 (46 sequences in the data set). Similarly, we did the same for 46 sequences of length 2. These are the Long Sequence Truncation and Short Sequence Truncation transformations. In order to check Sequence Length vs. Noisy Words, we added one contiguous token to the 46 sequences of length 4 (Long Sequence Extension) and to the 46 sequences of length 2 (Short Sequence Extension). As the table shows, only Int-F and SL-ICM are sensitive to the sequence length effect on captured or noisy tokens, but to a lesser extent than to previous transformations.

In order to check for Overlap Increasing Monotonicity, we produced three synthetic outputs, namely, -1 Tokens, -2 Tokens, and -3 Tokens, where 1, 2 or 3 correct tokens are removed from the sequence. We considered sequences that appeared alone in a sentence (532 affected sequences). According to the Overlap Increasing Monotonicity property, the score difference between the first and second outputs should be smaller than the difference between the second and third output. The only metric that satisfies this property is SL-ICM. In fact, the token-pair combination produces the opposite effect in link based metrics (i.e. link F1 and BC F1).

Similarly, in order to assess Noise Increasing Monotonicity, we generated +1 Token, +2 Tokens, and +3 Tokens transformations, where 1, 2 and 3 noisy tokens are added to the sequence. According to the Noise Increasing Monotonicity property, the score difference between +1 Token and +2 Tokens should be larger than between +2 Tokens and +3 Tokens. In this case, SL-ICM, Intersection based F-measure and BCubed metrics satisfy the property. In addition, Token-F and Link-F do not strictly satisfy the property, as they do not consider the sequence as an evaluation unit. However, the test is defined homogeneously for all sequence transformations and the property is empirically satisfied.

5 Conclusions

In this paper we have introduced a set of desirable formal properties when evaluating sequence labeling tasks. We have investigated which existing metrics satisfy each property and proposed a new metric (SL-ICM) based on information theory. Finally, we have conducted a series of experiments to assess how different metrics behave on different evaluation scenarios. Our theoretical and empirical results show that the new SL-ICM is the only metric compliant with all properties in all experiments.

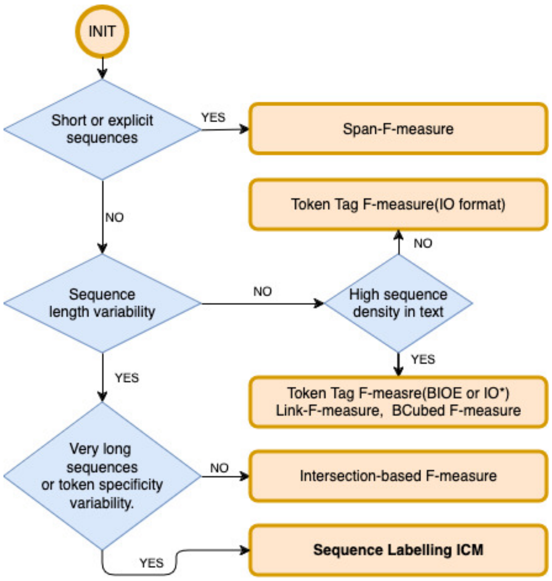
INIT
Short or explicit sequences
YES
Span-F-measure
NO
Token Tag F-measure(IO format)
NO
Sequence length variability
NO
High sequence density in text
YES
YES
Token Tag F-measre(BIOE or IO*) Link-F-measure, BCubed F-measure
Very long sequences or token specificity variability.
NO
Intersection-based F-measure
YES
Sequence Labelling ICM

Figure 2: Metric selection flow.

As a result of the analysis conducted in this paper, Figure 2 provides a guide to select the most suitable metric depending on the task and scenario. If exact matching between the system output and the gold standard is the main requisite, then span F-measure should be the metric of choice. This may occur in situations where the sequences are very short (one or two tokens per sequence). If the sequences are long and partial matching plays a role, then token F-measure with a simple annotation format (IO) may be appropriate, unless there are contiguous sequences. Then, it is better to use token F-measure with BIO or BIOE encoding, which consider the sequence boundaries. This is likely in applications such as text chunking. If there is a large variance in the length of the sequences, then we may want to give more weight to each token when the sequence is short, and it is more advisable to use a metric such as Intersection-based F-measure. This is the case of heterogeneous Named Entity Recognition, which includes short names and long descriptors. Finally, some sequence labeling scenarios include very long sequences (answer retrieval or quotation detection, for instance). Then we need to consider not only the length of the sequence but also the number of tokens that have been previously identified in the sequence, and SL-ICM should be the metric of choice. SL-ICM is also suitable when there is a high token specificity variability within sequences, for instance, stopwords in named entities.

6 Limitations

In this work we have introduced a set of formal properties that can be applied to sequence labeling evaluation metrics, and a novel metric for evaluating sequence labeling tasks that satisfies all the desired properties. The properties we have presented may help characterize metrics and get a better understanding of the evaluation results of a system.

These properties, however, may not be exhaustive, and other researchers may identify additional properties that may be relevant in certain scenarios. Similarly, the metric we have proposed can be useful in certain scenarios and tasks, but no evaluation metric is definitive or perfect for every case: the suitability of a metric is defined by the use case and our proposed metric should not be taken as the best suited for any usage scenario.

Another limitation is that the proposed metric satisfies all desirable formal properties identified in our formal analysis, but it comes at the cost of interpretability: compared with span exact matching, for instance, SL-ICM results are harder to interpret. There is usually a tradeoff between metric simplicity and metric adequacy, which may hinder adoption. We provide an open-source implementation of the metric to facilitate adoption[3], but interpretation will always be more challenging than span-based Precision and Recall.

Finally, our paper covers three of the four general methods proposed in the methodology taxonomy by Amigó et al. (2018): (i) Theoretical top-down, consisting of defining a priori properties to be satisfied by metrics, (ii) theoretical bottom-up, consisting of formally generalizing metrics such as span F-measure and Span Similarity, and (iii) empirical top-down, consisting of checking metrics on synthetic data generated from formal property example cases and transformations of the gold standard in a real data set. The fourth methodology, i.e., empirical bottom-up (which would consist of testing the metrics on real systems and data), is outside the scope of this article due to space limitations, but will be considered for future developments.

Acknowledgements

This work is partially funded by the Spanish Ministry of Science, Innovation and Universities (project FairTransNLP PID2021-124361OB-C32) funded by MCIN/AEI/10.13039/501100011033, by a research agreement between UNED and Red.es (C039/21-OT-AD2) and by ERDF, EU A way of making Europe. Elena Álvarez Mellado is funded by a FPI-UNED 2019 pre-doctoral grant.

References

Enrique Amigo and Agustín Delgado. 2022. Evaluating extreme hierarchical multi-label classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5809–5819, Dublin, Ireland. Association for Computational Linguistics.

Enrique Amigó, Hui Fang, Stefano Mizzaro, and ChengXiang Zhai. 2018. Are we on the right track? An examination of information retrieval methodologies. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 997–1000, New York, NY, USA. Association for Computing Machinery.

Enrique Amigó, Fernando Giner, Julio Gonzalo, and Felisa Verdejo. 2020. On the foundations of similarity in information access. *Information Retrieval Journal*, 23:216 – 254.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.

Parul Awasthy, Bishwaranjan Bhattacharjee, John Kender, and Radu Florian. 2020. Predictive model selection for transfer learning in sequence labeling tasks. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 113–118, Online. Association for Computational Linguistics.

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*.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-Grained Analysis of Propaganda in News Article. 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 5636–5646, Hong Kong, China. Association for Computational Linguistics.

[3] https://sites.google.com/view/enriqueamigo

Andrea Esuli and Fabrizio Sebastiani. 2010. Evaluating Information Extraction. In *Multilingual and Multimodal Information Access Evaluation*, Lecture Notes in Computer Science, pages 100–111, Berlin, Heidelberg. Springer.

Richard Johansson and Alessandro Moschitti. 2013. Relational features in fine-grained opinion analysis. *Computational Linguistics*, 39(3):473–509.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. MultiCoNER: A large-scale multilingual dataset for complex named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3798–3809, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the Second Shared Task on Language Identification in Code-Switched Data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas. Association for Computational Linguistics.

Nam Nguyen and Yunsong Guo. 2007. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 681–688, New York, NY, USA. Association for Computing Machinery.

Sean Papay, Roman Klinger, and Sebastian Padó. 2020. Dissecting Span Identification Tasks with Performance Prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4881–4895, Online. Association for Computational Linguistics.

John Pavlopoulos, Jeffrey Sorensen, Léo Laugier, and Ion Androutsopoulos. 2021. SemEval-2021 task 5: Toxic spans detection. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 59–69, Online. Association for Computational Linguistics.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Lev Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Adwait Ratnaparkhi. 1996. A Maximum Entropy Model for Part-Of-Speech Tagging. In *Conference on Empirical Methods in Natural Language Processing*.

Nathan Schneider, Emily Danchik, Chris Dyer, and Noah A Smith. 2014. Discriminative lexical semantic segmentation with gaps: running the mwe gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206.

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.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Supplementary Material

Appendix A: SL-ICM Derivation

SL-ICM is based on the Information Contrast Model similarity scheme (Amigó et al., 2020) which is a linear combination of the information content of the compared elements and their union:

$$ICM(A,B) = \alpha I(A) + \alpha_2 I(B) - \beta I(A,B)$$

In order to satisfy general similarity formal properties, the parameters must comply with $\alpha_1, \alpha_2 < \beta < \alpha_1 + \alpha_2$. We set the parameters $\alpha_1 = \alpha_2 = 2$ and $\beta = 3$.

$$\text{SL-ICM} = 2 \cdot I(\hat{S}) + 2 \cdot I(S) - 3 \cdot I(\hat{S}, S)$$

The motivation for these parameters is that they fit into the legal theoretic ranges. In addition, according to these parameters A system that does not provide any information is penalized based on the amount of information provided in the gold standard. A system that generates the same output as the gold standard is rewarded according to the amount of information in the gold standard.

$$I(\hat{S}) = 0 \Longrightarrow I(\hat{S}, S) = I(\hat{S}) = 0 \Longrightarrow \text{SQ-ICM} = -I(S)$$
$$\hat{S} = S \Longrightarrow I(\hat{S}) = I(S) = I(\hat{S}, S) \Longrightarrow \text{SQ-ICM} = I(S)$$

In the calculation of aggregate information it is necessary to discard redundant information. For example, if $\hat{S} = S$ then $I(\hat{S}, S) = I(\hat{S})$. On the contrary, if $\hat{S}$ and S are totally disjoint sequences, then we can sum their information content to quantify the joint information. In order to avoid redundant information, we approach the joint information quantity $I(\hat{S}, S)$ as the sum of the information quantities from the system output and the gold standard ($I(\hat{S})$ and $I(S)$), minus the amount of redundant information, which is the information content of the intersection between both sets of sequences.

$$I(\hat{S}, S) = I(\hat{S}) + I(S) - I(\hat{S} \cap S)$$

Leaving SL-ICM defined as:

$$\begin{aligned}
&\text{SL-ICM}(\hat{S}, S) = 2 \cdot I(\hat{S}) + 2 \cdot I(S) - 3 \cdot I(\hat{S}, S) \\
&= 2 \cdot I(\hat{S}) + 2 \cdot I(S) - 3(I(\hat{S}) + I(S) - I(\hat{S} \cap S)) \\
&= 3 \cdot I(\hat{S} \cap S) - I(\hat{S}) - I(S)
\end{aligned}$$

Computing intersection between $\hat{S} \cap S$ is not obvious. The intersection of sequences should have less information than the original ones. For this reason, it is necessary to map each sequence in system output with a corresponding sequence in the gold standard. For example, if the system output is one sequence $\hat{S} = \{\textit{John Mike}\}$ and the gold consists of two sequences, $S = \{\textit{John}, \textit{Smith}\}$, the intersection should be either $\hat{S} \cap S = \{\textit{John}\}$ or $\hat{S} \cap S = \{\textit{Smith}\}$, since otherwise, we would have more information in the intersection (two sequences) than in the original $\hat{S}$. The fundamental problem is that we cannot intersect a long sequence with multiple short ones, but only with one. SL-ICM takes the sequence intersections that maximize the information in both directions. $\hat{S}_t \vec{\cup} S_t$ represents the set of sequences in $\hat{S}_t$ intersected with the sequence in S_t with which it shares the most information. $S_t \vec{\cup} \hat{S}_t$ is the opposite:

$$\hat{S}_t \vec{\cup} S_t = \bigcup_{\hat{s} \in \hat{S}_t} \{s\} \cap \operatorname{argmax}_{s \in S_t} (I(\hat{s} \cap s, t))$$

$$S_t \vec{\cup} \hat{S}_t = \bigcup_{s \in S_t} \{s\} \cap \operatorname{argmax}_{\hat{s} \in \hat{S}_t} (I(s \cap \hat{s}, t))$$

where $I(s, t)$ represents the information quantity associated to labelling a sequence s as t. Then we consider the sequences that fit in both directions:

$$\hat{S}_t \cap S_t = \bigcup_t \left(S_t \vec{\cup} \hat{S}_t \cap \hat{S}_t \vec{\cup} S_t\right)$$

Figure 1 illustrates the sequence set intersection. Notice that we could find some counter samples for this approach. For example, it could be the case that, when many contiguous sequences of S and $\hat{S}$ cross each other like bricks, the bidirectional intersection never coincides and the intersection is lost. This could be solved by a more complex algorithmic process, but we believe that in sequence labeling these situations are too infrequent. Therefore, we have preferred a simpler option. For the sake of clarity, the figure does not include discontinuous sequences, but the behavior in that case would be similar.

Once $\hat{S}_t \cap S_t$ is solved, there are no overlapping sequences left. That is, no token belongs simultaneously to two sequences with the same tag. This fits in the intersection set $\hat{S}_t \cap S_t$, the system output set $\hat{S}_t$, and the gold standard S_t. Under this condition, the information of sequence sets can be quantified as the sum of their information contents, i.e. $I(\hat{S}_t) = \sum_{\hat{s} \in \hat{S}_t} I(\hat{s})$.

Now, the next issue is to compute the information content of assigning a tag t to the sequence s. We compute the information content of a sequence labelling (s, t) according to Shannon's information content definition ($I(x) = \log \frac{1}{P(x)}$). We assume that the probability of a sequence s to be tagged as t is the joint probability of the label and the word sequence. Assuming independence:

$$P(s, t) = P(t) \cdot P(s) \simeq \frac{N_t}{N} \cdot P(s)$$

And therefore:

$$I(s, t) = -log\left(\frac{N_t}{N}\right) + I(s)$$

where $N_t = \sum_{s \in S_t} |s|$, and $N = \sum_{s \in S_t} |s|$ represent the amount of tokens annotated as t in the goldstandard and the total amount of tokens respectively. The component $\frac{N_t}{N}$ makes the metric reward hits in infrequent tags.

According to the properties defined in this paper, the word information should decrease asymptotically with respect to the sequence length (OVERLAP INCREASING MONOTONICITY). We assume that the information contribution decreases according to $\frac{1}{l}$ being l the sequence length. That is,

$$I(\{w_1, \ldots, w_l\}) = I(\{w_1, \ldots, w_{l-1}\}) + \frac{1}{l} I(\{w_l\})$$

In terms of probability, this implies that the conditional probability of a token given the previous sequence is the l-th root of the isolated token probability:

$$\begin{aligned} & I(w_1, \ldots, w_{l-1}, w_l) \\ & = I(w_1, \ldots, w_{l-1}) + \frac{1}{l} I(w_l) \\ & = -log(P(w_1, \ldots, w_{l-1})) - \frac{1}{l} log(P(w_l)) \end{aligned}$$

Given that

$$I(w_1, \ldots, w_l) = P(w_l | w_1, \ldots, w_{l-1}) \cdot P(w_1, \ldots, w_{l-1}))$$

then

$$-log(P(w_l | w_1, \ldots, w_{l-1}))) = -\frac{1}{l} \log(P(w_l))$$

which implies that

$$P(w_l | w_1, \ldots, w_{l-1}) = P(w_l)^{\frac{1}{l}}$$

For example, if the probability of a token in a specific domain is $\frac{1}{100} = 0.01$, then it is assumed that the conditional probability of that token as the second element of a sequence is 0.1. After 10 tokens, it would be 0.63, providing much less information.

Therefore, being $s \in S_t$, the corresponding information quantity is:

$$I(s, t) = -log\left(\frac{N_t}{N} \prod_{i=1..l} P(w_i)^{\frac{1}{i}}\right) \tag{1}$$

If the single token probability is unkown, we can assume a constant probability $P(w) = k$ where k is less than one:

$$I(s, t) = -log\left(\frac{N_t}{N} \prod_{i=1..l} k^{\frac{1}{i}}\right)$$

Finally, being S, and $\hat{S}$ the sequences sets in the system output and the goldstandard, SL-ICM$(\hat{S}, S)$ can be expressed as:

$$\sum_{t \in T} \left(3 \cdot \sum_{s \in S_t \bigcap \hat{S}_t} I(s, t) - \sum_{s \in S_t} I(s, t) - \sum_{s \in \hat{S}_t} I(s, t)\right)$$

Appendix B:SL-ICM Properties

First, if S and $\hat{S}$ do not present partial matches, the SQ-ICM lead to the (non hierarchical) multilabel classification metric ICM (Amigo and Delgado, 2022). Being W the total set of words in the text and being $\hat{s}(w)$ and $s(w)$ the labels assigned to w by the system output and the gold standard respectively:

$$
\begin{aligned}
\text{SQ-ICM} = \sum_{w\in W} 3 &\left(\sum_{t\in\hat{s}(w)\cap s(w)} -\log\left(\frac{|N_t|}{|N|}\right)\right) \\
&- \sum_{t\in\hat{s}(w)} -\log\left(\frac{|N_t|}{|N|}\right) - \sum_{t\in s(w)} -\log\left(\frac{|N_t|}{|N|}\right) \\
&= \sum_{w\in W} \left(\sum_{t\in\hat{s}(w)\cap s(w)} -\log\left(\frac{|N_t|}{|N|}\right)\right) \\
&- \sum_{\substack{t\in\hat{s}(w)\setminus s(s)\\ t\in s(w)\setminus\hat{s}(s)}} -\log\left(\frac{|N_t|}{|N|}\right)
\end{aligned}
$$

satisfying Correct Sequence Monotonicity and Wrong Sequence Monotonicity. In addition, other classification oriented properties described in (Amigo and Delgado, 2022) are satisfied, such as True Category Specificity and Wrong Category Specificity.

In addition, according to Equation 1 the more a sequence is short, the more adding a new word to the sequence increases the information quantity, given that:

$$
\begin{aligned}
&-log\left(\frac{N_t}{N}\prod_{i=1..l} k^{\frac{1}{i}}\right) - \left(-log\left(\frac{N_t}{N}\prod_{i=1..l-1} k^{\frac{1}{i}}\right)\right) \\
&> -log\left(\frac{N_t}{N}\prod_{i=1..l+1} k^{\frac{1}{i}}\right) - \left(-log\left(\frac{N_t}{N}\prod_{i=1..l} k^{\frac{1}{i}}\right)\right)
\end{aligned}
$$

Therefore, the rest of properties are satisfied:

- Adding correctly labeled words to a sequence, increases $I(\hat{S}\cap S)$ and, to a lower extent, $I(\hat{S})$, but not $I(S)$, complying with Overlap Monotonicity.

- Adding wrong tokens to a sequence, increases $I(\hat{S})$ but not $I(\hat{S}\cap S)$ and $I(\hat{S})$, complying with Noise Monotonicity.

- When splitting a sequence incorrectly, $I(\hat{S}\cap S)$ decreases and $I(\hat{S})$ increases, satisfying Sequence Completedness.

- When joining incorrectly correct sequences into two contigous sequences, $I(\hat{S}\cap S)$ decreases to the same extent than $I(\hat{S})$, satisfying Sequence Homogeneity due to the metric parameters.

- The information content changes to a greater extent in short sequences when adding or removing words. Therefore, Sequence Length vs. Captured Words and Sequence Length vs. Noisy Words are satisfied.

- The information content changes to a greater extent in short sequences when adding or removing words. Therefore, Overlap Increasing Monotonicity are satisfied via the length of $I(\hat{S}\cap S)$ and $I(\hat{S})$. Noise Increasing Monotonicity are satisfied via the length of $I(\hat{S})$.