



UvA-DARE (Digital Academic Repository)

Bcubed revisited: elements like me

van Heusden, R.; Kamps, J.; Marx, M.

DOI

[10.1007/s10791-024-09436-7](https://doi.org/10.1007/s10791-024-09436-7)

Publication date

2024

Document Version

Final published version

Published in

Discover Computing

License

CC BY

[Link to publication](#)

Citation for published version (APA):

van Heusden, R., Kamps, J., & Marx, M. (2024). Bcubed revisited: elements like me. *Discover Computing*, 27(1), Article 5. <https://doi.org/10.1007/s10791-024-09436-7>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Research

Bcubed revisited: elements like me

Ruben van Heusden¹ · Jaap Kamps² · Maarten Marx¹

Received: 18 November 2022 / Accepted: 23 April 2024

Published online: 06 May 2024

© The Author(s) 2024 [OPEN](#)

Abstract

BCubed is a mathematically clean, elegant and intuitively well behaved external performance metric for clustering tasks. BCubed compares a predicted clustering to a known ground truth clustering through elementwise precision and recall scores. For each element, the predicted and ground truth clusters containing the element are compared, and the mean over all elements is taken. We argue that BCubed overestimates performance, for the intuitive reason that the clustering gets credit for putting an element into its own cluster. This is repaired, and we investigate the repaired version, called “Elements Like Me (ELM)”. We extensively evaluate ELM from both a theoretical and empirical perspective, and conclude that it retains all of its positive properties, and yields a minimum zero score when it should. Synthetic experiments show that ELM can produce different rankings of predicted clusterings when compared to BCubed, and that the ELM scores are distributed with lower mean and a larger variance than BCubed.

Keywords Information retrieval · BCubed · Clustering · Metrics

1 Introduction

We review the external clustering performance metric *BCubed* [3], indicate a flaw and propose a repair. We then evaluate the repair both theoretically and experimentally.

In essence, clustering and (single label) classification perform the same task: given a set of items E , they partition E . However, when it comes to evaluation with comparison to a gold standard, things are very different.

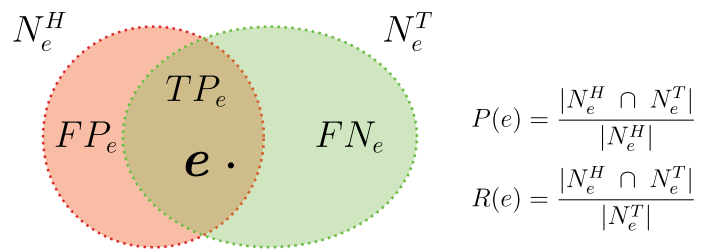
With classification, the number of blocks in the partition is known (the set of labels), and a mapping exists between the true blocks and the predicted blocks (namely the identity mapping on the labels). So, counting errors is straightforward by making the cross table of predicted and gold truth values (the *confusion table*), and computing precision and recall as the diagonal divided by the two margins, respectively.

This is an extension of BCubed Revisited: Elements Like Me, published in ICTIR 2022. This work has several key differences and extensions over the previous work. First, theoretical proofs on the behaviour of ELM with respect to BCubed have been added in Sect. 3.1, 3.2 and 3.3 with proofs of the ZeroScore constraint, behaviour on degenerate clusters and proof of its ability to change rankings. A new Sect. 3.4 has been added that shows the differences between BCubed and ELM on synthetically generated data where rankings between all possible clusterings off 14 elements were compared to investigate the ranking differences between BCubed and ELM. A new experiment has been conducted, where the original hierarchical clustering algorithm is replaced with a BERT model (Sect. 3.5). The literature section has been updated to include an overview of the applications of BCubed. Apart from these large differences, several more minor changes were also made. Figure 1 was added to illustrate the BCubed intuition, an impression of the differences using synthetic data (Sect. 2.2) was added, and several plots were improved. We estimate 30–50% of this work is new or significantly updated over the original paper.

✉ Ruben van Heusden, r.j.vanheusden@uva.nl; Jaap Kamps, kamps@uva.nl; Maarten Marx, m.j.marx@uva.nl | ¹IRLab, Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands. ²ILLC, Faculty of Humanities, University of Amsterdam, Amsterdam, The Netherlands.



Fig. 1 Comparing the elements in the true cluster N_e^T of e to those in the predicted cluster N_e^H of e . TP_e , FP_e , and FN_e represent the sets of True Positives, False Positives, and False Negatives for e , respectively. $P(e)$ and $R(e)$ are Precision and Recall relative to e



With clustering, there is (at prediction time) no known number of blocks (as the label set is unknown), and there is no mapping between the predicted blocks and the true labels. This makes counting errors much less straightforward, witnessed by the numerous proposals on how to do this, nicely surveyed and classified by Amigó, Gonzalo, Artiles, and Verdejo [1].

The BCubed measure, proposed by Bagga and Baldwin [3], sidesteps the problem of matching true and hypothesized clusters. It does not measure errors over the clusters, but computes a precision and recall value for each element, and then takes the average. i.e., the recall for element e is the fraction of the true cluster of e that is contained in the predicted cluster of e . As each element e is contained in both its true and predicted cluster, both recall and precision of e are always larger than 0, even when a predicted and true clustering are disjoint except for the element e . This can be repaired by leaving out e itself in the calculation of precision and recall of e . In this paper, we investigate this alternative definition of BCubed (Sect. 2), evaluate the new metric both theoretically and empirically (Sect. 3), and conclude that it retains all positive properties of BCubed, yields a minimum zero score when it should and can produce different rankings for predicted clusterings when compared to BCubed.

2 BCubed revisited

Let E be a set and N^T and N^H two clusterings (partitions) of E , corresponding to the *true* and *hypothesized* clustering, respectively. We use N_e^T to denote the block in N^T containing e , and similarly for N_e^H and N^H . Figure 1 shows how precision and recall relative to an element e are defined given the true and hypothesized clusterings N^T and N^H .

The BCubed measure for a given clustering is then the average harmonic mean (the F1-value) of the precision and recall for each element. This F1 value is what is denoted by “BCubed” or “BCubed score” in the literature, a convention also followed in this paper. This harmonic mean is usually defined as $2PR/(P + R)$, but the equivalent direct definition is insightful here as well. Let $A \oplus B$ denote the symmetric difference of the sets A and B . Then

$$F1(e) = \frac{|N_e^H \cap N_e^T|}{|N_e^H \cap N_e^T| + .5 \cdot |N_e^H \oplus N_e^T|} \tag{1}$$

Fig. 1 shows that $TP_e \neq \emptyset$, as e is always in TP_e and thus that precision, recall and F1 are always positive for each element, implying that the BCubed score of a clustering is never equal to 0.

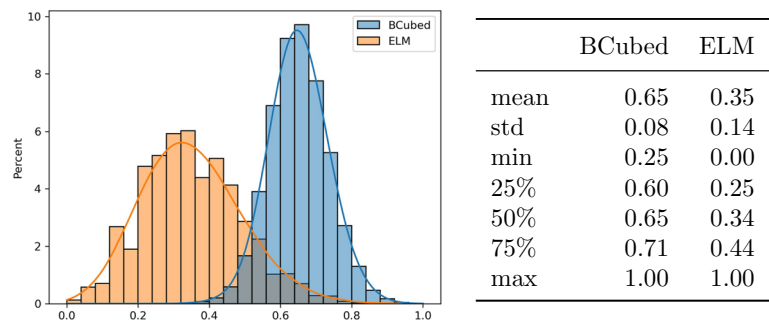
Having a meaningful zero point is a requisite for a metric to be measured on ratio-scale. We can say that a score of 0 is meaningful if none of the predictions was correct, thus when all items in the contingency table are off the diagonal. Let us formulate this as a desideratum for a clustering metric:

(ZeroScore) For every true clustering, there is a predicted clustering with score 0.

BCubed fails the ZeroScore constraint and can even give quite a high score of .66 to an absolutely wrong prediction. Consider this simple example: $E = \{1, 2\}$ and the true clustering N^T is $\{E\}$. Now let the predicted clustering N^H be (the only other possibility) $\{\{1\}, \{2\}\}$. Obviously it is wrong, but for both elements e , $P(e) = 1$, as it makes no mistakes, $R(e) = .5$, as half of the true elements of the block of e are in its predicted block, and so $F1(e) = .66$. Taking the mean F1 over all elements, we get a BCubed score of .66 for this predicted partition.

In fact, because TP_e can never be empty, BCubed fails the ZeroScore constraint in a much stronger manner: for every true clustering there is no prediction with score 0. Bcubed can never be equal to 0.

Fig. 2 Population density diagram of all BCubed and ELM scores for all 16,384 predictions of the model in (2) plus the main statistics



In order to repair BCubed so that it does satisfy the ZeroScore constraint, we only need to remove e from both N_e^H and N_e^T . Thus N_e^T now denotes the set of all elements in the same true cluster as e except e itself, and similarly for N_e^H . We call these the *neighbors* of e . Then $TP_e = N_e^H \cap N_e^T$ can be empty, and thus all measures can be equal to 0. The price paid for this is that we may divide by 0 in the definitions of P , R and $F1$ and thus must account for that. So all definitions remain the same, but we add the following provisos:

- If $N_e^H = \emptyset$, $P(e) = 1$.
- If $N_e^T = \emptyset$, $R(e) = 1$.
- If $N_e^T = N_e^H = \emptyset$, $F1(e) = 1$.

With these rules the new definitions yield the same scores as the original BCubed definitions on the singleton cases. In the first case, the hypothesized cluster containing e is $\{e\}$, thus no mistakes for e can be made. In the second, recall for e is indeed perfect, and a perfect F1 score for a true singleton is of course only obtained if we exactly predict that.

It is easy to see that with this proviso the definition of $F1(e)$ as in (1) is still equivalent to the often used $2PR/(P + R)$ formulation.

2.1 A new name

In the rest of the paper, we further evaluate this repair. But let us first give it a name. The BCubed measure was introduced by Bagga and Baldwin. In a footnote they attribute the idea of BCubed to Bierman, and thus the cubed Bs. We opted for *ELM*, an abbreviation of *Elements Like Me*, which is a good mnemonic of the way we compute the repaired BCubed measure.

2.2 First impression of the differences

The following example gives a good impression of the difference between the two measures. Let E be the set consisting of the first 15 digits, and let it have the following true clustering

$$\{\{1, 2\}, \{3, 4, 5\}, \{6, 7\}, \{8\}, \{9\}, \{10, 11, 12\}, \{13, 14\}, \{15\}\}. \quad (2)$$

We have generated all possible predictions with the proviso that each cluster must consist of consecutive elements. For a set of N consecutive elements, there are 2^{N-1} of these. For 15 elements, this results in 16,384 possible predicted partitions. Figure 2 shows the distribution of the BCubed and ELM scores for all these predictions. Both scores are approximately normally distributed and 25% of the ELM scores are below the lowest BCubed score. The ELM scores are more evenly spread over the possible scores. Not only does BCubed start higher, its variance of .006 is much lower than the .02 for ELM. ELM and BCubed can also rank clustering systems differently: in this example, 18% of all ($2^{14} \times 2^{13}$) pairs of predictions are ranked differently by ELM and BCubed.

The Kendall-Tau β statistic (which accounts for tied ranks) between the ELM- and the BCubed-based ranking in this example is 0.63, also indicating that there are substantial differences between the rankings produced by the two metrics.

3 Evaluation

We evaluate the new ELM metric both theoretically and empirically in a number of ways:

1. Unlike BCubed, ELM satisfies the ZeroScore constraint.
2. ELM has good behaviour on extreme non-informed (referred to as “degenerate” by Beeferman, Berger, and Lafferty [5]) systems.
3. The ranking of clustering systems based on ELM can be different from the ranking based on BCubed.
4. There are systematic differences between ELM and BCubed in synthetic experiments.
5. There are systematic differences on a real dataset, with a state-of-the-art clustering system based on BERT.
6. ELM satisfies the same four clustering metric constraints developed by Amigó et al. [1] that BCubed satisfies.

We compare the BCubed and ELM versions of P , R and $F1$ using superscripts $P_B, P_{ELM}, R_B, R_{ELM}, F1_B$ and $F1_{ELM}$. In the proofs, the precision, recall and $F1$ scores over a set of elements E are defined as the average of these scores over all elements in E .

3.1 ZeroScore constraint

Theorem 1 *For every set E , with at least 2 elements, and a gold standard partition N_T over E , there is a clustering N_H over E such that for every e in E , $F1_{ELM}(e) = 0$.*

Proof Let E and N^T be as stated in the Theorem. Let $E_s \subseteq E$ be the set of elements which are clustered into singletons. There are three cases: no true singleton clusters, exactly one true singleton cluster or more than one singleton cluster. If there are no true singleton clusters, simply let the predicted clustering partition E into singletons. Recall that we now use N_e^H and N_e^T as denoting all elements in the same cluster as e except e . In particular, with a true singleton cluster $\{e\}$, $N_e^T = \emptyset$. Then for each $e \in E$, $N_e^H = \emptyset$ and $|N_e^T| \geq 1$ (as $|E| \geq 2$). And thus $TP_e = \emptyset$ and $F1(e) = 0$, because the special clause for $F1$ does not apply. If there is more than one true singleton cluster, create the predicted clustering N^H as follows: one cluster E_s and for each $e \in E \setminus E_s$, a singleton cluster $\{e\}$. Again, we must show that $F1(e) = 0$, for each $e \in E$. First, let $e \in E_s$. Then $N_e^T = \emptyset$ and $N_e^H = E_s \setminus \{e\}$, which is not equal to \emptyset as E has at least 2 elements. And thus $TP_e = \emptyset$ and $F1(e) = 0$ because the special clause for $F1$ does not apply. If $e \notin E_s$, the reasoning is as in the case without singletons. If E_s is itself a singleton, say $\{s\}$, we proceed as follows. Because E has at least two elements, it has another element different from s , say t . Let N^H consist of the cluster $\{s, t\}$ and, again, for each $e \in E \setminus \{s, t\}$, a singleton cluster $\{e\}$. Using the same argument as above, for each $e \in E$, $TP_e = \emptyset$ and $F1(e) = 0$, as the special case never applies. \square

3.2 ELM behaves well on degenerate clusterings

Theorem 2 *Let N^T be a true clustering over a set E and N^H the clustering consisting only of singleton clusters. Then $P(e) = 1$, for all $e \in E$, and $R(e) = F1(e) = 1$ only if $N_e^T = \emptyset$ and 0 otherwise.*

An immediate corollary is that the ELM $F1$ for the degenerate singleton clustering is equal to the proportion of singletons in the gold standard partition.

Proof Assume E , N^T and N^H are as in the theorem. In particular then $N_e^H = \emptyset$, for all $e \in E$. Then by the special clause in the definition, $P(e) = 1$ for all e , and $R(e) = F1(e) = 1$ if $N_e^T = \emptyset$. When $N_e^T \neq \emptyset$, still $N_e^T \cap N_e^H = N_H(e) = \emptyset$, and thus both $R(e)$ and $F1(e)$ are 0. \square

Now consider the other degenerate clustering: all elements are contained in one cluster. Let N^H be this degenerate all in one predicted clustering, with N^T the true clustering over a set E . Then obviously, $R(e) = 1$, for all $e \in E$. Because $N_e^H = E \setminus \{e\}$ and thus $N_e^T \cap N_e^H = N_e^T$ the precision $P(e)$ equals $\frac{|N_e^T|}{|E|} - 1$. And thus the mean precision equals

Table 1 F1 scores per element and the mean, for the given true and two system clusterings over the set $\{1, 2, 3, 4, 5\}$, according to both BCubed and ELM

Metric	System	1	2	3	4	5	Mean
BCubed	H_1	$\frac{2}{3}$	$\frac{2}{3}$	1	1	1	0.87
BCubed	H_2	1	1	$\frac{4}{5}$	$\frac{4}{5}$	$\frac{1}{2}$	0.82
ELM	H_1	0	0	1	1	1	0.60
ELM	H_2	1	1	$\frac{2}{3}$	$\frac{2}{3}$	0	0.66

$$P = \frac{\sum_{e \in E} |N_e^T|}{|E| \cdot (|E| - 1)} = \frac{\sum_{c \in N^T} |c| \cdot (|c| - 1)}{|E| \cdot (|E| - 1)} = \frac{\sum_{c \in N^T} |c|^2 - |c|}{|E|^2 - |E|},$$

where the $c \in N^T$ denote the true clusters. Note that the BCubed mean precision for this degenerate clustering is equal to $\frac{\sum_{c \in N^T} |c|^2}{|E|^2}$. Also note that when we view the clustering as a directed network partitioned into cliques, the ELM precision equals the *density* of this network, which ranges from 0 when each clique is a singleton to 1 only if the network is complete and thus consists of one giant cluster.

We can conclude that for both degenerate clusterings, ELM gives the lowest reasonable score.

3.3 ELM can produce different rankings compared to BCubed

We give an example of a true clustering and two predicted clusterings (which can be seen as two competing systems), which are ranked differently by ELM compared to BCubed. The clusterings are over the set $E = \{1, 2, 3, 4, 5\}$ and are given in the first 3 rows of Table 1, with for example H_1 , depicting the clustering $\{1\}, \{2\}, \{3, 4, 5\}$. The other rows compute $F1(e)$ for each element, for each clustering and using ELM and BCubed. System H_2 is better according to ELM, while H_1 is better according to BCubed.

Both H_1 and H_2 contain one error, but the error in H_2 is in the larger cluster. Clustering intuition says that errors in smaller clusters should be penalized more than errors in larger ones, and that is what ELM does here, and BCubed does not.

3.4 ELM vs BCubed on synthetic data

We expand on the small synthetic experiment conducted in Sect. 2.2 by computing the BCubed and ELM scores for all clusterings of size 14 against all other clusterings of size 14. As there is a total of 2^{14-1} possible clusterings, we thus have 8,192 experiments, with each of these experiments producing two rankings of the predicted clusters, one for BCubed and one for ELM. The distribution of these scores is shown in Fig. 3. To further investigate the differences between BCubed and ELM when used to rank systems, we calculate the Kendall-Tau statistic between all rankings and also look at the number of system pairs where the order was swapped between BCubed and ELM (which is part of the calculation of Kendall-Tau). The number of pairs where the ranking order was swapped between BCubed and ELM was roughly 39 billion out of the 274 billion cases (14%). The Kendall-Tau over all pairs of rankings is normally distributed with a mean of 0.70 and a standard deviation of 0.06, also similar to the example in Sect. 2.2.

Figure 4 shows the distribution of the fraction of the number of swaps for all of the 8,192 experiments. Thus each datapoint is the fraction of possible system pairs where the order between the ranking between BCubed and ELM was swapped for that particular ranking. The y-axis represents all 8192 rankings and indicates what percentage of all rankings has a certain fraction of swaps. To investigate which type of ground truth clusterings result in the largest number of

Fig. 3 Population density diagram of BCubed and ELM scores between all possible pairs of ground truth and predicted clusterings of size 14 plus the main statistics ($N = 2^{13} \cdot 2^{12}$)

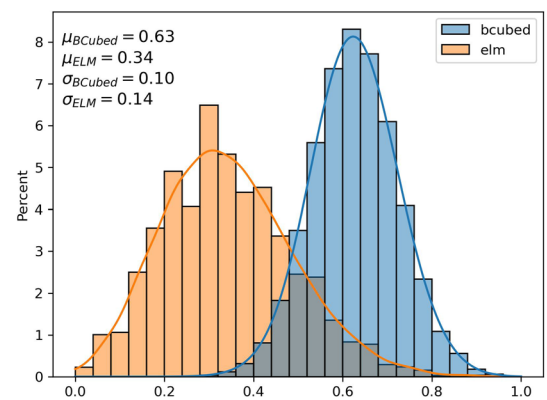
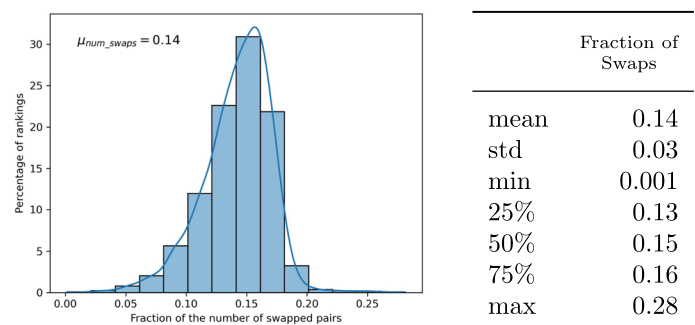


Fig. 4 The fraction of the number of pairs with reversed orderings between BCubed and ELM for all rankings for all clusterings with size 14 ($N = 8192$) plus the main statistics



swaps, we employ the Pearson correlation between the entropy of the ground truth clustering and the number of pairs swapped in the ranking. The entropy for a given ground truth clustering N^T is given by

$$entropy(N^T) = - \sum_{C \in N^T} p(C) * \log_2(p(C)), \text{ where } p(C) = \frac{|C|}{|E|}.$$

The Pearson correlation between the entropy of a ground truth clustering and the number of swaps for that ground truth was 0.81 ($N = 8192$). As clusterings that have more small elements have higher entropy, this means that the number of swaps tends to be higher when the ground truth has a larger number of small clusters. This is as expected as the difference between ELM and BCubed is larger on smaller clusters.

3.5 ELM vs BCubed on real data

We compare ELM and BCubed for three fixed cluster-size baselines and a clustering algorithm using BERT [7] on a large dataset consisting of 110 samples (separate clustering problems) with in total 24,180 true clusters over in total 89,491 elements. The mean and median cluster sizes are 4 and 2, respectively. Each sample is a sequence of pages of text divided into documents. Thus each cluster consists of a document, which is a continuous sequence of pages. The elements are thus the pages. This scenario is common in the field of Page Stream Segmentation [20]. On average roughly 35% of the clusters in a stream are singleton clusters.

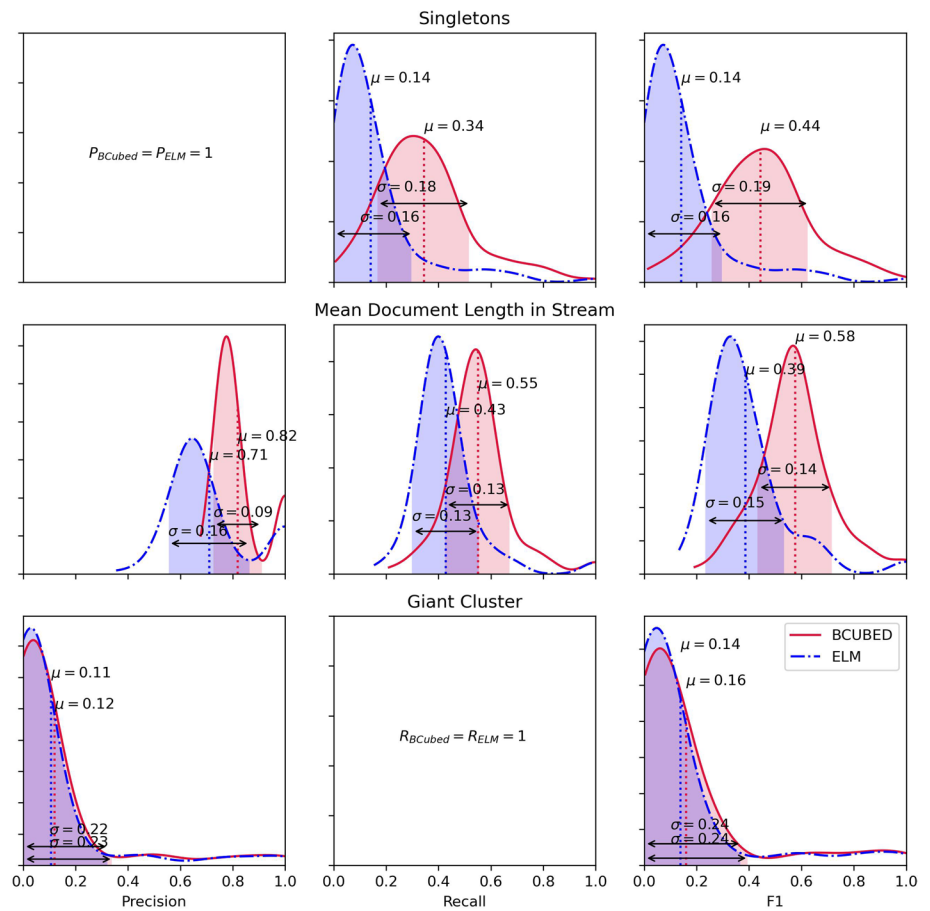
Following Bagga and Baldwin [3] and Amigó et al. [1], we report the mean average $F1$ scores. Thus for every sample E in our testset, we take the average over the $F1(e)$ for each $e \in E$, and then we take the mean over all samples in the testset.

The dataset, together with the code for all the experiments conducted in this paper is available on GitHub.¹ For the experiments with the BERT model, the dataset was split into a 70% train and 30% test set.

As the fixed page size baselines are not learned, we can use the entire dataset (train and test) for these experiments. The BERT model is evaluated on the test part of the dataset, consisting of 34 samples with 6347 clusters over 25,676 elements.

¹ <https://github.com/irlabamsterdam/elm>.

Fig. 5 Distributions of mean average Precision, Recall and F1 for BCubed and ELM for the three fixed baselines (only singletons, one giant cluster, and each cluster has the length of the samples mean true cluster length ($N = 110$))



We first compare the precision, recall and their harmonic mean for BCubed and ELM on three fixed baselines: the two degenerate clusterings with only singleton clusters and one giant cluster, and a more sensible baseline that evenly partitions a sample into clusters of the mean true cluster size of that sample. The results are shown in Fig. 5. Note how the plots for precision for the all-singleton prediction and recall for the one-giant-cluster prediction show constant values of 1 for both ELM and BCubed.

The plots indicate that the smaller the cluster sizes in the predicted clustering, the larger the difference in both the mean and standard deviation of BCubed and ELM, for all 3 measures, again as expected.

We will now cluster this dataset using the BERT model for Page Stream Segmentation from Guha, Alahmadi, Samanta, Khan, and Alahmadi [9]. In short, this model creates textual representations of each page using a BERT model, and then uses this representation to divide the pages into pages starting a new document and other pages. This classification is equivalent to a clustering. We follow their experimental setup, replacing the English *bert-base* model with the Dutch version² as the dataset is in Dutch. We train the model for 10 epochs, using a batch size of 512 and a learning rate of $2e^{-5}$.

Table 2 shows hardly any difference in precision and recall, but still a three percent point difference in F1 score. The KDE plots of the differences in Fig. 6 show the same trend.

This result shows that the differences between ELM and BCubed do not only exist on synthetic and simple baseline models, but also on well performing state-of-the-art models tested on large real data.

² <https://huggingface.co/GroNLP/bert-base-dutch-cased>.

Table 2 Mean BCubed and ELM precision, recall and F1 scores for the BERT based clustering model evaluated on the Page Stream Segmentation dataset (N = 34)

	Precision	Recall	F1
BCubed	$\mu = 0.93, \sigma = 0.07$	$\mu = 0.85, \sigma = 0.26$	$\mu = 0.83, \sigma = 0.24$
ELM	$\mu = 0.92, \sigma = 0.08$	$\mu = 0.85, \sigma = 0.26$	$\mu = 0.80, \sigma = 0.24$

Fig. 6 KDE Plots of the relative differences between BCubed and ELM for Precision, Recall and F1 for the BERT clustering model (N = 34)

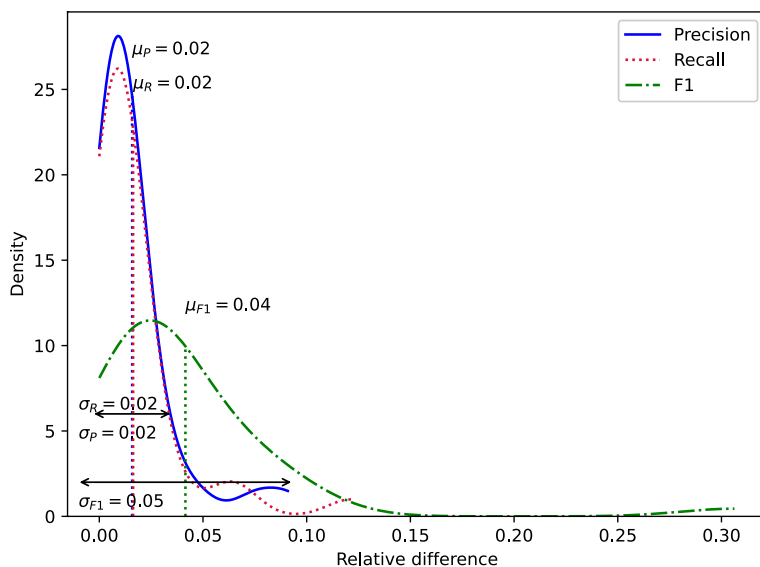
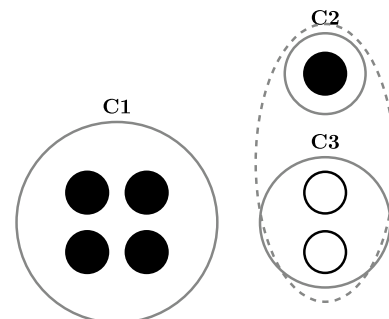


Fig. 7 Homogeneity constraint: black nodes belong to one cluster and the white nodes belonging to another cluster. Shown are two partitions: the homogeneous $D_1 : \{C_1, C_2, C_3\}$ and the mixed $D_2 : \{C_1, C_2 \cup C_3\}$. Figure 7 is a modification of Figure 5 from [1]



3.6 ELM satisfies the constraints of Amigó et al

We show that the four constraints developed by Amigó et al. [1] hold for the ELM F1 metric. The family of BCubed-like cluster evaluation metrics is the only one satisfying all these four constraints. For a thorough explanation and motivation of these constraints we refer to the original paper. We follow the same line of reasoning as Amigó et al. [1] and also use their informative pictures.

3.6.1 Homogeneity

The homogeneity constraint states that a cluster assignment D_1 that splits samples into homogeneous subgroups should be scored higher than an assignment D_2 that mixes samples of different subgroups together, like in Fig. 7.

The ELM recall for each element is the same in D_1 and D_2 , but the precision is lower for the elements in the mixed cluster in D_2 , than in the homogeneous clusters in D_1 . Hence, the mean ELM F1 score of D_1 is higher.

Fig. 8 Completeness constraint: All nodes belong to the same cluster shown are two partitions: $D_1 = \{C_1, C_2 \cup C_3\}$ and $D_2 = \{C_1, C_2, C_3\}$. Figure 8 is a modification of Figure 6 from [1]

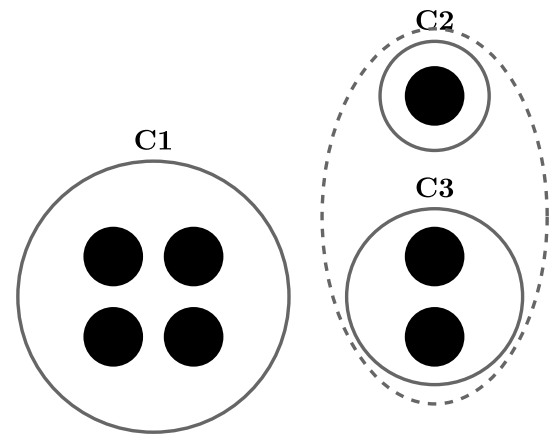
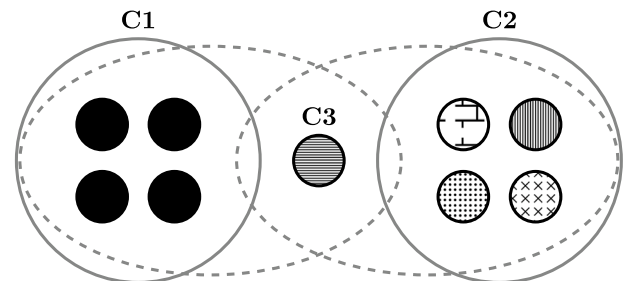


Fig. 9 Rag Bag constraint: black nodes belong to one cluster and all other nodes are singleton clusters shown are two cluster assignments: $D_1 = \{C_1, C_2 \cup C_3\}$ and $D_2 = \{C_1 \cup C_3, C_2\}$. Figure 9 is taken from Figure 7 from [1]



3.6.2 Completeness

The cluster completeness constraint states that a cluster assignment D_1 that groups items belonging to the same cluster together should receive a higher score than a clustering D_2 that subdivides items from a homogeneous cluster, like in Fig. 8.

The argument is the dual of the previous argument. Here, precision is maximal for all elements in both partitions as all clusters are homogeneous. But ELM recall is lowered for those elements in the separate C_2 and C_3 . In fact, recall for ELM is 0 for singleton clusters. Thus the mean ELM $F1$ is higher for the partition D_1 with the joined clusters.

3.6.3 Rag Bag

The Rag Bag constraint states that adding a singleton cluster to a cluster consisting of all differently labeled elements, a *rag-bag*, should score higher than an assignment adding this singleton to a homogeneous cluster, as in Fig. 9. In this example, this means that D_1 should score higher than D_2 .

First observe that all elements have the same recall in both clusterings. Now the element in C_3 has the same precision of 0 when it is added to C_1 or to C_2 . The elements in the rag-bag C_2 also keep the same precision (namely 0) irrespective to whether C_3 is joined or not. But those in the homogeneous C_1 see a drop in precision (from 1 to $\frac{3}{4}$) when C_3 is joined. Thus D_1 has a higher mean ELM $F1$.

3.6.4 Cluster size vs. quantity

As stated by Amigo et al., the Cluster Size vs. Quantity constraint can be loosely formulated by saying that small mistakes in large clusters should be penalized less than small mistakes in small clusters. Amigo et al. operationalize this constraint as follows. Let $n > 2$, and E a set of elements with $|E| = 3n + 1$, and let T, H_1 and H_2 be three partitions over E , where T is the ground truth, and H_1 and H_2 are two predicted clusterings. Let T be a partitioning of E containing one cluster C_1 of size $n + 1$, and n clusters each of size 2, C_2 through C_{n+1} . Let H_1 be a partitioning of E that splits C_1 into a cluster C'_1 of size n , and C''_1 of size 1, and with C_2 through C_{n+1} unaltered. Let H_2 be a partitioning that leaves C_1 unaltered, but splits C_2 through C_{n+1}

Fig. 10 An illustration of the Cluster Size Vs. Quantity constraint for ELM for $n = 3$ and $E = \{1, 2, \dots, 10\}$. The numbers in the two bottom rows are the ELM F1 scores for each element, and the mean F1 (the ELM score)

T											
H ₁											
H ₂											
	1	2	3	4	5	6	7	8	9	10	Mean
H ₁	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	0	1	1	1	1	1	1	$\frac{8}{10}$
H ₂	1	1	1	1	0	0	0	0	0	0	$\frac{4}{10}$

into $2n$ singleton clusters $\{C_2^L, C_2^R, \dots, C_{n+1}^L, C_{n+1}^R\}$. An illustration of this setup for $n = 3$ is given in Fig. 10. The thus formalized constraint now says that the ELM score of H_1 should be higher than that of H_2 , given T .

Theorem 3 (Cluster Size Vs. Quantity) *Given $n > 2$, T, H_1 and H_2 as described above, the ELM F1 score for H_1 is higher than that for H_2 .*

Proof Let T, H_1 and H_2 be as stated in the constraint for some $n > 2$. Given that both H_1 and H_2 only split true clusters in T into smaller subsets, $P(e) = 1$ for every element in E for both H_1 and H_2 , and thus proving that the mean ELM F1 is larger for H_1 than for H_2 simplifies to proving that this holds for the mean recall. We will show that the sum of all $R(e)$ is higher for H_1 than for H_2 , which proves the theorem.

For H_1 , the recall of all $2n$ nodes belonging to the correctly predicted clusters C_2 through C_{n+1} equals 1, and the recall of the single node in C_1'' is 0 (this would be $\frac{1}{n+1}$ for BCubed). The ELM recall of all n nodes in C_1' equals $\frac{n-1}{n}$ (this would be $\frac{n}{n+1}$ for BCubed). Thus for H_1 , $\sum_{e \in E} R(e)$ equals $2n + n \cdot \frac{n-1}{n} = 3n - 1$.

For H_2 (which correctly predicts the big cluster but splits all true two-size clusters) the ELM recall $R(e) = 0$, for all $e \in C_i$ with $i \neq 1$ (this would be $\frac{1}{2}$ for BCubed). For the $n + 1$ nodes in the correctly predicted C_1 the recall is 1, and thus for H_2 , $\sum_{e \in E} R(e) = n + 1$. For every $n > 1$, $3n - 1 > n + 1$, as desired. □

4 BCubed in the literature

We survey for which tasks BCubed has been used and discuss two other refinements of BCubed.

BCubed is used in the Machine Learning community for several clustering problems where a gold standard clustering is available, such as coreference resolution [6, 14, 15, 17, 18], Entity Linking [10, 11], and name disambiguation [2, 8]. In the case of coreference resolution, the task is to map words or short phrases that occur in a text to real-world entities. This mapping defines a clustering of all these words and phrases.

In coreference resolution in particular, BCubed is often used as a successor to the link based metric used in MUC [19]. BCubed has two main advantages over MUC: its ability to score singleton clusters, and the fact that it takes the severity of clustering mistakes into account, something MUC does not. ELM obviously still keeps these advantages. In both coreference resolution and Entity Linking, cluster size is likely long tail distributed, with a few very large clusters and numerous smaller clusters, and many singletons. We have seen that BCubed especially overestimates on elements from small clusters and that ELM repairs this. As the reported F1 measure is the mean over all elements, this skewed distribution amplifies the overestimation. We thus believe that especially in these applications, ELM is preferable to BCubed.

Several refinements of BCubed have been proposed, to adapt the metric to specific use-cases. Moreno and Dias [13] proposed two adjustments to the BCubed F1 metric that makes it more suited for usage with highly unbalanced datasets, which for example occur frequently in the tasks of image clustering, or the clustering of results for ambiguous search terms on the web. They argue that the standard version of BCubed is less suited for this, because the larger clusters (of the irrelevant class) have an unreasonable effect on the total score, comparable to the unreasonableness of accuracy in such cases. Both proposed alterations have the effect of weighting precision more than recall. The most straightforward one is not to use the harmonic mean F1, but a differently weighted average. The same remedy can be applied to ELM by using different weights in equation (1) for FP_e and FN_e .

An extension to BCubed that handles overlapping clusters correctly is proposed by Amigó et al. [1], where the quality of a predicted cluster is evaluated by comparing an element with all other elements (including itself) in the ground truth (for recall, predicted cluster for precision) and comparing how many clusters they share in the prediction compared to the ground truth. However, this extension might assign the maximum F1 score to a clustering that is not exactly equal to the gold standard. Rosales-Méndez and Ramírez-Cruz [16] propose *CICE-BCubed*, which fixes the aforementioned issue for BCubed by also checking for pair occurrences in different classes. The adapted BCubed variant proposed by Amigo et al. that makes it suitable for usage with overlapping clusterings (and the change proposed by the authors of CICE-BCubed), is not straightforward to implement for ELM. The main problem arises from the fact that this extended variant of BCubed must include a comparison between the element and itself, to be able to penalize a model for the spurious creation or deletion of singleton clusters. Consider the example where the ground truth contains two elements e_1 and e_2 that both belong to cluster a , and a prediction where e_1 and e_2 both belong to a , but e_1 also belongs to a new cluster b . Intuitively, the precision for this element should not be 1, as the prediction added a cluster, but the definition of ELM means that this relation is not considered, and thus this mistake is not penalized. We leave the repair of this shortcoming of ELM in the case of overlapping clusters for future work.

5 Discussion

We have calculated the F1 scores for both BCubed and ELM on the element level, and then defined the F1 score of a predicted clustering as the average of the F1 scores of all elements. Although we believe this is closest to the original (not explicitly stated) definition as given by Bagga and Baldwin [3], this is not the only way in which BCubed can be defined. Amigó et al. [1] define BCubed from the average precision and recall over all elements and then applying the $2PR/(P + R)$ manner of calculating the F1 score using these averages. In words: we have used the average harmonic mean instead of the harmonic mean of the averages. For the main message of this article this does not matter as both ways of defining BCubed do not satisfy the ZeroScore constraint.

6 Conclusion

We indicated that the BCubed F1 measure gives an overestimation of the performance of a clustering method, repaired the definition, and evaluated the result positively.

ELM satisfies a basic property of a metric: it can always obtain the minimal score of 0, and it gives it to each prediction which has nothing correct (i.e., not a single true positive). We want to emphasize that the idea and intuition behind the ELM metric is identical to that of BCubed.

We showed that the difference between ELM and BCubed is largest when the size of true clusters is small and when there are many of such small clusters (e.g. when cluster size is power law distributed). Even on large real datasets with a well performing state-of-the-art clustering algorithm, ELM F1 was three percent point lower than BCubed.

We end with looking at the problem from the perspective of network science [4, 12]. If we view a clustering not as a set of subsets on some domain D but as a *binary relation on D* , we take a network perspective. A clustering or partition then corresponds to an equivalence relation \equiv . The neighbor function $N(e) = \{e' \in D \mid e \equiv e'\}$ then is the clustering function used to define BCubed and ELM. In network science, it is customary to work with simple (that is, irreflexive), and if possible, undirected relations. If we replace the equivalence relation with this irreflexive undirected relation, we end up with the same partition (in network science the blocks are called *cliques*). But on this network, the same neighbor function defines ELM, simply because no element is a neighbor of itself. We may speculate how BCubed would have been defined if one of the three B's had been a network scientist.

Acknowledgements We sincerely thank the reviewers of the manuscript for their detailed and constructive feedback and suggestions, which helped to improve the overall quality of the paper.

Author contributions Conceptualization, M.M., J.K. and R.H.; Resources, M.M and R.H.; Data Curation, R.H; Software R.H and M.M.; Formal Analysis, M.M and R.H; Supervision, M.M and J.K.; Funding Acquisition, J.K. and M.M.; Validation, R.H and M.M.; Investigation, R.H and M.M.; Visualization, R.H and M.M.; Methodology, R.H, J.K. and M.M.; Writing—Original Draft, M.M and R.J.; Project Administration, R.H and M.M.; Writing—Review and Editing, all authors.

Funding This research was supported in part by the Netherlands Organization for Scientific Research (NWO) through the ACCESS project grant CISC.CC.016, and by the University of Amsterdam through Humane AI.

Data availability Both the data and the source code used in this research are publicly accessible on GitHub via the following link: <https://github.com/irlabamsterdam/elm>

Declarations

Competing interests The authors have no conflict of interest to declare that are relevant to the content of this article.

Ethics approval Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Amigó E, Gonzalo J, Artiles J, Verdejo F. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf Retr J*. 2009;12(4):461–86.
2. Artiles J, Borthwick A, Gonzalo J, Sekine S, Amigó E. Weps-3 evaluation campaign: overview of the web people search clustering and attribute extraction tasks. *CLEF (notebook papers/labs/workshops) 2010*;1176.
3. Bagga A, Baldwin B. Entity-based cross-document coreferencing using the vector space model. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*. 1998;1:79–85.
4. Barabási A-L, Pósfai M. *Network science*. Cambridge: Cambridge University Press. Retrieved from <http://barabasi.com/networkscience-book/> 2016.
5. Beeferman D, Berger A, Lafferty J. Statistical models for text segmentation. *Mach Learn J*. 1999;34(1):177–210.
6. Beheshti S-M-R, Benatallah B, Venugopal S, Ryu SH, Motahari-Nezhad HR, Wang W. A systematic review and comparative analysis of cross-document coreference resolution methods and tools. *Comput J*. 2017;99(4):313–49.
7. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pretraining of deep bidirectional transformers for language understanding. *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)* (pp. 4171–4186). 2019.
8. Ferreira AA, Gonçalves MA, Laender AH. A brief survey of automatic methods for author name disambiguation. *ACM SIGMOD Rec*. 2012;41(2):15–26.
9. Guha A, Alahmadi A, Samanta D, Khan MZ, Alahmadi AH. A multi-modal approach to digital document stream segmentation for title insurance domain. *IEEE Access*. 2022;10:11341–53.
10. Ji H, Grishman R, Dang HT, Griffitt K, Ellis J. Overview of the TAC 2010 knowledge base population track. *Proceedings of the Third Text Analysis Conference (2010)* (Vol. 3, pp. 3–3). 2010.
11. Ji H, Nothman J, Hachey Bea. Overview of tac-kbp2014 entity discovery and linking tasks. *Proceedings of the Text Analysis Conference (2014)* 2014;1333–1339.
12. Menczer F, Fortunato S, Davis CA. *A first course in network science*. Cambridge: Cambridge University Press; 2020. <https://doi.org/10.1017/9781108653947>.
13. Moreno JG, Dias G. Adapted B-CUBED metrics to unbalanced datasets. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval 2015*;p. 911–914.
14. Poot C, van Cranenburgh A. A benchmark of rule-based and neural coreference resolution in Dutch novels and news. *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference 2020*;79–90.
15. Rahman A, Ng V. Supervised models for coreference resolution. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing 2009*;968–977.
16. Rosales-Méndez H, Ramírez-Cruz Y. CICE-BCubed: a new evaluation measure for overlapping clustering algorithms. *J. Ruiz-Shulcloper & G. Sanniti di Baja (Eds.), Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (pp. 157–164). Springer.2013.
17. Stylianou N, Vlahavas I. A neural entity coreference resolution review. *Expert Syst with Appl*. 2021;168: 114466.
18. Uzuner O, Bodnari A, Shen S, Forbush T, Pestian J, South BR. Evaluating the state of the art in coreference resolution for electronic medical records. *J Am Med Inf Assoc*. 2012;19(5):786–91.
19. Vilain M, Burger JD, Aberdeen J, Connolly D, Hirschman L. A model-theoretic coreference soring scheme. *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 1995*;6-8, 1995.
20. Wiedemann G, Heyer G. Multi-modal page stream segmentation with convolutional neural networks. *Lang Resour Eval J*. 2021;55(1):127–50.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.