Establishing Annotation Quality in Multi-Label Annotations

Marian Marchal and **Merel Scholman** and **Frances Yung** and **Vera Demberg**
Language Science and Technology / Saarland University
{marchal, m.c.j.scholman, frances, vera}@coli.uni-saarland.de

Abstract

In many linguistic fields requiring annotated data, multiple interpretations of a single item are possible. Multi-label annotations more accurately reflect this possibility. However, allowing for multi-label annotations also affects the chance that two coders agree with each other. Calculating inter-coder agreement for multi-label datasets is therefore not trivial. In the current contribution, we evaluate different metrics for calculating agreement on multi-label annotations: agreement on the intersection of annotated labels, an augmented version of Cohen's Kappa, and precision, recall and F1. We propose a bootstrapping method to obtain chance agreement for each measure, which allows us to obtain an adjusted agreement coefficient that is more interpretable. We demonstrate how various measures affect estimates of agreement on simulated datasets and present a case study of discourse relation annotations. We also show how the proportion of double labels, and the entropy of the label distribution, influences the measures outlined above and how a bootstrapped adjusted agreement can make agreement measures more comparable across datasets in multi-label scenarios.

1 Introduction

Annotation efforts have long been characterized by the (implicit or explicit) assumption that there is a single true interpretation for every item, and all other interpretations are incorrect. This has become even more pervasive with the increasing usage of annotations as input for classifiers and other downstream computational tasks. However, the single-truth assumption has been challenged in recent years (e.g., Aroyo and Welty, 2015; Basile et al., 2021), based on counterexamples from many different fields showing that two coders can disagree on an annotation and still both be right, due to the subjectivity and complexity of many tasks. We here focus on the field of discourse relation annotation as a case study, although the insights could be applied to different types of linguistic research (Amidei et al., 2019; Uma et al., 2021).

Establishing inter-annotator agreement in scenario's where multiple labels are possible is challenging. In cases where multiple interpretations are possible but coders are restricted to annotate only a single label, it is unclear whether disagreement reflects incorrect interpretations (coder error or issues with the coding scheme or training) or item ambiguity. Given an instance where two interpretations are equally likely, assuming that at least one coder inferred both possible readings but only a single label can be annotated, the chance that the coders agree would be only 50%. This does not reflect that there is actually high agreement on the labels.

Allowing for multi-label coding can result in more accurate annotations as well as higher agreement, and with that most likely higher reliability of a final label. Although two sets of multi-labels can be compared by calculating the difference between their distributions (e.g. comparing the probability distribution of a classifier to the distribution of crowdsourced labels, Fornaciari et al., 2021), the distributions and ranks of the labels are not necessarily relevant or accessible (e.g. expert annotations where multiple labels are allowed).

We here assume a scenario where unranked multiple labels are allowed, regardless of whether more labels are possible and aim to properly estimate agreement. We will therefore focus on estimating the reliability of a single final label per item, while also discussing other scenario's. Traditional agreement statistics, such as Cohen's Kappa (Cohen, 1960) will not be suitable for evaluating the reliability of data that allow for multi-label annotations – at least not without adjustments. It should be taken into account that multi-label coding also inflates the chance agreement: by providing more labels, there is a higher chance that at least one of those labels overlaps with the annotations from another

Proceedings of the 29th International Conference on Computational Linguistics, pages 3659–3668
October 12–17, 2022.

coder.

The current paper will analyze several ways to determine the reliability of multi-label annotations. We argue that chance agreement should always be taken into account for any agreement measure to make it comparable across datasets. The contributions of this paper are: (i) we propose a bootstrapping method to estimate the expected agreement (see Section 4), and (ii) we compare agreement on various measures for simulated datasets with different parameters, as well as for a real-life dataset (Section 5 and 6). We make available a reproducible script of the implementations of the measures and the calculations.[1]

We use the following terminology throughout the paper. An *item* (i) is an instance which is annotated. It can be annotated with one or more *labels*, which together make up one *annotation*. *Categories* (k) refer to the options that *coders* (c) have to assign an item to.

2 Multiple labels: Ambiguity and uncertainty

There are several scenarios in which multiple labels are possibly desirable. A first one is a scenario in which annotators are uncertain which of multiple categories are correct, caused by a lack of knowledge or information (Beck et al., 2020). Allowing them to provide multiple labels would then reflect a probabilistic representation of the target label. These probabilistic labels also occur often in crowd-sourced annotations, where workers might lack the knowledge to select the correct label.

Secondly, multiple labels might reflect true ambiguity. These are cases where more than one interpretation is possible. To illustrate, consider the discourse relation in Example (2): both a SPECIFICATION relation (which could be expressed by inserting the cue phrase *more specifically* between the two sentences), and a MANNER relation (which could be expressed by *to do so,*) interpretation are valid.

(1) Ryan was decorating the Christmas tree. He was hanging the baubles.

Ambiguity has been argued to be an "inherent property of natural language" and an important source of disagreement in language annotation (Beck et al., 2020). Plank et al. (2014) show that the vast majority of disagreements between annotators arise because multiple labels are valid and recommend allowing for such disagreements, rather than focusing on inter-annotator agreement. The present contribution shows that it is possible to establish inter-annotator agreement while taking ambiguity into account by allowing for multiple labels. More specifically, the focus of the present paper is on establishing a reliability measure for ambiguous cases where multiple labels are valid, but we will also address how reliability measures can reflect these different multi-label scenarios.

3 Reliability measures

Reliability is the extent to which different coders arrive at the same interpretations of items. Reliability can be measured by calculating the inter-coder agreement using an agreement coefficient: a numerical index of the extent of agreement between the coders. However, the goal of obtaining reliable data is not merely to have data on which two coders agree (coders might be wrong or biased, after all), but to have annotated labels that reflect the true meaning of the items. It is important to note that this validity, despite being the goal of annotation efforts, is not captured by agreement coefficients.

Agreement coefficients usually consist of two components: observed agreement (A_o) and expected agreement (A_e). Together, these can be used to calculate an adjusted agreement, i.e. an agreement coefficient (A_c):

$$A_c = \frac{A_o - A_e}{1 - A_e} \quad (1)$$

Observed agreement is taken to be 1 when two coders assign an item to the same category and 0 when the item is assigned to different categories. Observed agreement does not take into account chance agreement, which occurs when one or both coders rate an item randomly. In order to get a reliable index of the extent of agreement between coders, observed agreement therefore has to be adjusted for the proportion of agreement that is expected to occur by chance. The crucial difference between various inter-coder agreement measures often lies in the way in which they estimate this expected agreement (see Artstein and Poesio, 2008 for a detailed overview).

One of the most frequently used inter-coder agreement measures is Cohen's Kappa (κ) (Co-

[1] https://osf.io/f5v4p/?view_only=4962b8ae4398466c88e620c27302e0c5

hen, 1960). When each data point in a corpus is assigned a single label, calculating chance agreement, and κ, is straightforward. More specifically, Cohen's κ calculates the probability of each label being selected by each coder independently:

$$A_e = \sum_{k \in K} P(k \mid c_1) \cdot P(k \mid c_2) \quad (2)$$

However, traditional kappa is not applicable to multi-label scenarios. One of the simplest solutions would be to treat each multiple-label annotation as a distinct label, but this inflates the number of categories in the coding scheme, which negatively affects κ. The traditional κ measure would therefore need to be adapted to make it suitable for multi-label annotations.

3.1 Soft-match agreement

One solution to adapt the traditional kappa measure is to calculate agreement using the intersection of agreed-upon labels – that is, the label that occurs in the annotation of both coders. For example, in Table 1, this leads to the observed match agreement being 1 for each item.

Taking the intersection agreement as the final annotation increases the probability that the obtained label is part of the set of true labels for that item, because both annotators agree on that label. Discarding the additional, non-overlapping labels means that information regarding that item is lost, but for many tasks, such as analyses for psycholinguistic experiments, or training certain classifiers, researchers only use a single label. In such scenarios, discarding the additional labels does not negatively impact the results.

To calculate `soft-match` agreement, multi-label annotations of the coders are replaced with the intersecting label (e.g., as done in Crible and Degand, 2019). In cases where there is no intersection or two intersecting labels between two multi-label annotations, a single label is sampled in order to be able to estimate expected agreement. For example, in item 1 in Table 1, κ will be calculated after removing label B from c_2 in item 1 and sampling either A or B for both coders for item 3. κ is then calculated on this adjusted dataset as usual, using the formulas above. This type of agreement can also be considered the *oracle agreement*, as it is the highest agreement that coders could have reached if they had selected only the overlapping single label.

Table 1: Example items with observed agreement for soft-match (S), augmented kappa (A), recall (R, c_1 wrt c_2) precision (P, c_1 wrt c_2) and F1.

	annotations		**observed agreement**				
item	c_1	c_2	S	A	R	P	F1
1	A	A;B	1	.50	1	.50	.67
2	A;B	B;C	1	.25	.50	.50	.50
3	A;B	A;B	1	.50	1	1	1

However, this method of calculating kappa on the intersection is problematic because it does not take into account that chance agreement on an intersection is higher when multiple labels are provided, over-estimating κ. In the most extreme case, where one coder would assign all categories to a single item, both observed and expected agreement will be 1 in reality. However, expected agreement using this `soft-match` agreement is much lower, thus inflating κ. Kappa on the `soft-match` agreement can therefore be misleading.

3.2 Augmented kappa

Rosenberg and Binkowski (2004) proposed an augmented version of κ, referred to here as `augmented kappa`, to measure corpus reliability for multi-labeled instances. In their approach, multiple labels are considered not as distinct selections, but as one divided selection, with a probability distribution over the different labels. Thus, it reflects a scenario where annotators are uncertain about which label is correct. For `augmented kappa`, each label for a specific item receives a weight which equals 1 divided by the number of labels annotated to that item.[2] For example, in item 1, label A for c_1 receives the full weight, as in a single label scenario. For c_2, label A as well as label B get a weight of 0.5. The observed agreement of an item i is then defined by:

$$A_o^i = \sum_{k' \in k'_i} W_{c1}^{k'} W_{c2}^{k'} \quad (3)$$

where k'_i is the set of intersecting labels of item i and $W_{c1}^{k'}$ and $W_{c2}^{k'}$ are the weights of the label annotated by each coder. The overall observed agreement (A_o) is the mean value of the observed agreement per item (A_o^i). The expected probability for coder c to annotate the category k for a dataset with n items is defined as:

[2]The original approach accommodates assigning different weights to each label for an item, so a distinction can be made between primary and secondary labels.

$$P(k|c) = \frac{1}{n} \sum_{i \in I} W_c^k \quad (4)$$

and the overall expected agreement is calculated as in Formula 2. For the items in Table 1, expected agreement according to this measure would be .39.

Conceptually, this measure equals the κ that would have been obtained if coders were only allowed to give one label and had randomly selected one of the multiple labels they provided. To illustrate, in the second item in Table 1, randomly selecting the labels from the two coders would result in agreement in 25% of the cases.

However, this `augmented` measure always penalizes providing multiple labels, because multiple labels reflect coder uncertainty. As a result, this measure does not take into account that there is possibly true ambiguity, and that both labels provided by a single coder may be correct. Agreement can never be higher than when these labels have been selected by chance. More specifically, if both coders assign an item to the same multiple categories, as in item 3 in Table 1, observed agreement will be 0.5 according to this measure. This is not suitable for scenarios in which researchers want to account for the fact that the multiple labels arise from the fact that there might be more than one true label.

3.3 Precision, recall and F1

Another possible solution would be to consider metrics typical for evaluating computational approaches to annotation: `precision`, `recall` and `F1` (see Brants, 2000, for a similar approach). In annotation, this can be phrased in terms of the proportion of intersecting labels compared to the total set of labels provided by the first coder (`precision`) and by the second coder (`recall`). These measures are particularly useful when one of the labels serves as the gold, for example when quality of aggregated crowd labels is compared with a gold label standard, or when a new annotator trains with a more experienced annotator.

`Precision` and `recall` allow for multiple identical labels, while correcting for providing more labels than those that are agreed upon. For example, if one coder provides more labels than the other coder, as in item 1 in Table 1, observed `F1` is decreased. However, the traditional versions of these measures do not take into account what chance performance would be. Since chance agreement depends on a variety of factors, such as the number of available categories, observed `precision`, `recall` and `F1` are not comparable across datasets that vary in the amount of labels per item.

4 Bootstrapping expected agreement

The existing measures are problematic for calculating multi-label agreement, because they do not correct for chance agreement for multiple labels (`soft-match` and `F1`) or they penalize multiple labels even if both annotators agree on the double labels (`augmented kappa`). The main contribution of the current paper is therefore to suggest a bootstrapping method for obtaining chance agreement, that can be used to adjust existing measures.

We propose to sample from the provided distribution per coder, in order to estimate the true expected agreement needed to calculate the adjusted agreement. More specifically, for each item we draw from the distribution of labels provided by each coder, following Cohen's κ. This distribution is obtained by dividing the number of times each category has been assigned by the total number of labels provided. The amount of sampled labels is likewise sampled from the probability distribution of the amount of labels for each item occurring in the original data set. In other words, if a coder provided a single label in 40% of the items and two labels in 60% of the items, double labels are also sampled in 60% of the items in the simulated dataset.

By simulating the sample data, we can bootstrap the expected agreement of several measures. For example, we can calculate the average proportion of intersecting labels across *n* simulated datasets. Using the expected agreement obtained by the simulations, existing agreement measures can be adjusted by taking into account chance agreements, using Equation 1. This allows us to obtain a variety of new measures. For calculating agreement on the intersecting labels, similar to `soft-match`, we refer to this measure as `boot-match`. Its expected agreement is estimated using the bootstrapping method, contrasting it with the `soft-match` measure where the expected agreement is calculated after removing additional labels that are not an intersecting label (see Section 3.1). Similarly, bootstrapping the expected agreement allows us to correct `precision`, `recall` and `F1` for chance agreement. The traditional version of these mea-

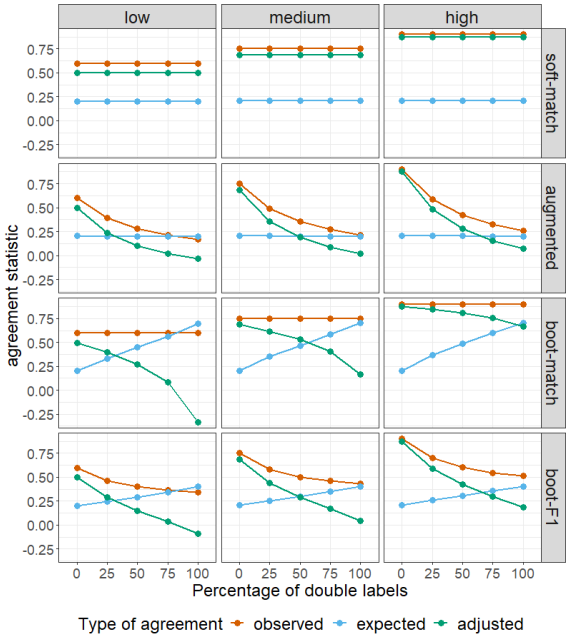

Figure 1: Agreement statistics per measure across different datasets. Each panel row displays one of the measures discussed in Sections 3 and 4. The agreement statistics on the y-axis are shown for simulated datasets with various parameters: Each panel column differs in the percentage of intersection agreement (low = 60%, medium = 75%, high = 90%) and the percentage of double labels can be found on the x-axis.

sures will be referred to as the observed components and the bootstrapping method allows us to also obtain the expected and chance-adjusted agreement components for these measures: We will therefore refer to these measures as `boot-F1`, `boot-precision` and `boot-recall`.

5 Comparing reliability measures on multi-label annotations

The previous sections show that there are various ways to estimate agreement in multi-label annotation tasks. The goal of an agreement measure is to be able to compare how much coders agree across different datasets. To evaluate which measure would be preferred for estimating agreement in multi-label datasets, a good measure should therefore (a) estimate expected agreement proportional to the amount of multi-labels and (b) provide a higher agreement score for tasks with more agreement. We will explore how the measures discussed above behave in different scenarios below.

To illustrate how the proportion of double labels as well as intersection agreement influence the scores, we simulate datasets with different characteristics for two parameters: the percentage of double labels provided and the percentage of observed intersection agreement. We thus manipulate how often a double label is chosen, ranging from never to always, with 25% intervals. In addition, the datasets vary in how much coders agree on the labels, which is manipulated by simulating data with various degrees of intersection agreement: *low* equals 60% intersection agreement, *medium* 75% and *high* 90%. For each of these parameter combinations, we sample 100 datasets. For each of these, we bootstrapped the expected agreement separately, using 100 simulations. Each dataset contained 100 items, which were assigned to one (or two) of five categories with equal probability. We then calculate the average observed, expected and adjusted (i.e. κ) agreement for each measure across these datasets.[3]

The agreement statistics for all the different datasets in the simulation analysis are provided in Figure 1. For the scenarios where no double labels are provided, the statistics are the same across the different measures: the adjusted agreement is 0.50, 0.69 and 0.87 for the different levels of intersection agreement. However, when more double-annotated labels are added, the patterns for the observed, expected and adjusted agreement diverge for the different measures.

The first row in Figure 1 considers *oracle* agreement, using the `soft-match` measure. Needless to say, the observed agreement (in red) increases when agreement is higher. Moreover, observed agreement for the `soft-match` measure remains constant when the percentage of double labels increases. However, obtaining intersection agreement is easier when more labels are provided. `Soft-match` does not take this into account, because the expected agreement does not change when more labels are provided. Across all simulated datasets, expected agreement is 0.21. As a result, the adjusted agreement for this measure is also constant across datasets with varying amounts of double labels. The adjusted agreement for `soft-match` therefore remains very close to the observed agreement, which is too liberal.

`Augmented kappa` corrects for multi-label scenarios. The expected agreement remains constant with this measure, regardless of the number

[3]Expected agreement for the subset of items where `soft-match` does not determine a single label, was calculated using the same method as for the expected agreement for `augmented kappa`.

of double labels. The observed agreement, on the other hand, decreases significantly. Note that this is partly due to the fact that the second label was sampled randomly. However, even if there would be perfect agreement on a double label for this measure (i.e. both coders assign an item to category A and B), the observed agreement is only 0.5 (see Table 1). In addition, observed agreement (and thus also the adjusted agreement) drops considerably when only 25% of items contain double labels. As a result, achieving a reliable adjusted agreement score is almost impossible with this measure: even when the intersection agreement is 90% and only a quarter of the items contain double labels, the adjusted agreement is still only 0.49. Finally, we point out that the adjusted agreement barely increases when the agreement is higher.

The observed agreement for the `boot-match` is the same as for `soft-match`, as it also considers intersection agreement. Like `soft-match`, it thus also remains constant when the dataset contains more double labels. Unlike the `soft-match` measure, however, the `boot-match` measure takes the percentage of double labels into account when calculating the adjusted agreement, as it estimates the expected agreement by simulating the provided distributions in the dataset 100 times. Expected agreement therefore increases when more labels are added, resulting in a lower adjusted agreement. With a fully double-label annotated dataset, expected agreement is 70%. In the low agreement scenario, the adjusted agreement is therefore even negative when all labels receive a double annotation. Using this adjusted agreement, we can compare how agreement on double-label datasets relates to datasets without double labels. For example, with five categories, achieving 90% intersection agreement with all double labels is comparable to achieving 75% agreement with 0% double labels (adjusted agreement ≈ 0.67). Note that this relationship depends on several parameters, such as the entropy of the label distribution and the number of categories. We will return to this issue below.

For the `boot-F1` measure, the observed agreement decreases when the percentage of double labels increases[4], similarly to `augmented kappa`. This is partly due to the fact that the manipulation of agreement was only for the intersection agreement and any additional labels besides the intersecting labels were sampled randomly. As a result, perfect agreement is not always achieved in the simulations, unlike for the `soft-match` and `boot-match` agreement. In this scenario, however, a perfect `F1` would be obtained if both coders assign an item to the same two categories. `Boot-F1` is slightly higher than `augmented`, across the different measures in the various datasets, because it yields higher agreement when both coders provide the same two labels (as in item 3 in Table 1).

More importantly, note that the expected `boot-F1` increases when more double labels are added to the dataset. Reporting only observed `F1` is therefore misleading, because achieving the same observed `F1` on a dataset with few double labels compared to one with many double labels is more difficult. The results are therefore not comparable across datasets. Calculating the chance-adjusted `boot-F1` solves this problem, because it takes this chance agreement into account.

5.1 Number of categories

These simulations reflect an annotation task with five categories. When more categories are added, however, expected agreement decreases, resulting in a higher adjusted agreement given the same observed agreement for all measures. In a zero double-label scenario, the adjusted agreement increases 5 percentage points when the number of categories increases from 5 to 10. For the measures for which expected agreement increases when more items have double labels, this increase is weaker with more categories. For example, for five categories, expected `boot-match` agreement increases from 0.20 (when no double labels are used) to 0.70 (when only double labels are used). With ten categories, this is an increase of 0.11 to 0.38. As a result, for a fully double-annotated dataset, obtaining 75% intersection agreement yields an adjusted agreement of 0.60 when ten categories are used, rather than $A_c = 0.17$ in a task with five categories. To conclude, the number of categories greatly affects the agreement statistics and should be kept in mind when evaluating the results of any annotation effort, e.g. by calculating an adjusted agreement rather than observed agreement only. Our proposed method of bootstrapping expected

[4] `Boot-recall` and `boot-precision` behave very similar to `boot-F1` here because the data is sampled similarly for the two coders. In the no and fully double-label scenarios they are exactly the same as the F1, in the in-between cases they are often slightly higher than the `boot-F1`.

agreement takes this into account.

5.2 Entropy

The entropy of the probability distribution of the labels also changes the results. The entropy reflects the likelihood that each label is chosen, and as such also influences the probability of agreement on the label:

$$H = -\sum p_k \, log \, p_k \tag{5}$$

As the categories in our simulation study are sampled with equal probability, the present distribution has a relatively high entropy. This leads to a relatively low chance agreement. The probability distribution can also have a lower entropy, if some categories are more prevalent in the dataset. For example, in the case of discourse annotations, `reason` relations occur more often in natural data than `contrast` relations. A probability distribution with a lower entropy results in a considerably higher expected agreement across all measures. Unlike with an equal probability distribution, the expected agreement for `augmented` and `soft-match` is not constant, but decreases slightly when more labels are added in a lower entropy scenario. For `soft-match` this means that the adjusted agreement even increases with more labels. Moreover, the expected agreement for `soft-match` is now much higher than that of the `augmented kappa`, because `soft-match` only calculates expected agreement after removing additional labels from items on which intersection is reached.

Finally, the decrease in the adjusted agreement for `boot-F1` and `boot-match` with more double labels is reduced. In real-world datasets, labels likely do not have equal probability, such as in our simulation analysis. Because the entropy of the probability distribution influences chance agreement, the adjusted agreement of the measure rather than the observed agreement should therefore be reported. This makes annotation agreement more comparable across datasets.

6 Case study

Real-world datasets often have very different characteristics than the simulated datasets in the previous section. We therefore explore how the measures behave in a real-world dataset: a case study for discourse relation annotation, which is a notoriously difficult task. It is often difficult to achieve a $\kappa > .7$ on single-label annotations (Spooren and Degand, 2010). This is partly due to the fact that discourse relation frameworks often distinguish many different categories (as can be seen below). In addition, coherence is not a feature of the text, but of the mental representations that readers have of the text (Sanders et al., 1992). Therefore, discourse relation annotations depend on coders' interpretation of the text, which may be subjective. Furthermore, ambiguity plays an important role for discourse relation annotation, which can partly explain low agreement with single label annotations. Recent studies have therefore turned to crowd-sourcing to source discourse relation annotations (e.g. Yung et al., 2019; Scholman et al., 2022; Pyatkin et al., 2020). This allows researchers to capture a larger variety of interpretations per instance. However, in some cases, such an approach is not suitable, or researchers still target a single final label.

The case study was part of a psycho-linguistic experiment in which participants were asked to provide a one-sentence continuation to a prompt. Two expert coders annotated the continuations with respect to their discourse relation with the prompt. The coding scheme allowed for maximally two labels, when coders believed both senses held. In total, the coders annotated 884 items, using 19 categories. 11 of these categories occurred in the final intersection label. The first coder had provided double labels in 11.4% of the cases, the second coder in 65.5% of the items. As a result, the first coder has a greater influence on the intersection item, because only one of the items by the second coder will be selected when one of them overlaps with the label provided by the first coder.

As can be seen in Table 2, agreement is moderate. The adjusted agreement of `soft-match` reflects the best-case scenario: if both coders only chose the single intersection label, the κ would have been .73. `Boot-match` corrects the expected agreement based on the proportion of double labels, resulting in a lower adjusted agreement.[5] `Soft-match` thus over-estimates agreement. According to `augmented kappa`, the adjusted agreement would be highly insufficient, because many double labels have been provided. This measure assumes that the multiple labels reflect coder uncertainty, but these expert annotators only provided two labels if they thought that both

[5]For the boot-strapped estimates presented in this section, we used 1000 simulations.

	observed	expected	adjusted
soft-match	.79	.23	.73
augmented	.51	.21	.38
boot-match	.79	.34	.68
boot-rec.	.55	.21	.43
boot-prec.	.76	.31	.65
boot-F1	.62	.24	.50

Table 2: Agreement statistics per measure for the case study (rec. = recall, prec. = precision).

labels were true. Boot-precision of c_2 with respect to c_1 is lower than boot-recall in this same direction. This can be attributed to the fact that the first coder provided fewer double labels than the second coder. Boot-F1 for this dataset is relatively low, even though the coders score high on the boot-match agreement. The reliability of the single final label is thus quite high, but the coders diverged on when and what additional labels should be provided.

Traditional F1 does not always reflect agreement properly, for two reasons. First, observed F1 will decrease when expert annotators find additional labels that might also be true. Secondly, observed F1 does not take into account the chance agreement on this measure. Adjusted boot-F1 and boot-match display agreement more accurately. As shown above, each measure provides different insights into the data quality and which measure(s) should be reported therefore depends on the goal of the annotation effort. Finally, the case study shows that even when annotators are instructed in the same way, the number of double labels that they provide still diverges between the two coders. Recall and precision provide more insight in this.

7 Related work

Bhowmick et al. (2008) also propose an adjusted κ measure to account for multi-labeled annotations. Crucially, their proposed metric considers the non-inclusion in a category by an annotator pair as an agreement. Such an approach is not optimal for annotation scenarios which can be characterized by a large number of categories in the coding scheme. This includes certain discourse relation annotation efforts, for which coding schemes can contain over 40 categories. With such large schemes, coders likely do not consider every category separately during annotation of a single item, but rather consider a subset of categories that seem most applicable to the item.

Finally, relating to the issue of ambiguity in annotation and selecting a single final label, we note that this is a debatable issue in itself. In correspondence to the assumption that items can express more than one meaning, a soft label – consisting of a probability distribution of all categories per item – more accurately captures an item's ambiguity. For example, CrowdTruth (Aroyo and Welty, 2013; Dumitrache et al., 2018) evaluates data quality by capturing the ambiguity inherent in semantic annotation through the use of disagreement-aware metrics. Fornaciari et al. (2021) and Uma et al. (2021) showed that models trained on soft labels, such as these, outperform those trained on single-label data, especially if they are evaluated using soft labels as well. A larger number of coders would be needed to more accurately calculate a probability distribution for an item. This is not the case for tasks that require (or choose to use) expert annotations only, as is the case in psycho-linguistics, or when annotating a gold dataset.

8 Discussion and conclusion

Annotating data with multiple labels better reflects the true meaning of the items, as these items can be ambiguous or even have multiple interpretations. After all, the goal of agreement measures is not to establish how strongly the coders agree, but rather how reliable the label is. The label on which annotators agree is more likely to be a true label, regardless of whether all the labels are captured. Obtaining a distribution of labels may be helpful in some, but not all, tasks (cf. Fornaciari et al., 2021). In addition, for many tasks such distributions are not available, either because not enough observations were obtained, or because computational models predict a limited number of labels. The present study therefore explored measures for evaluating various agreement measures on scenario's with more than one label.

The augmented kappa has been proposed as a measure of agreement on multi-label annotated datasets (Rosenberg and Binkowski, 2004), but it penalizes additional labels heavily and does not consider items assigned to the same multiple categories as full agreement. The underlying assumption is thus that there is only one true label, reflecting uncertainty rather than ambiguity. However, it is not always true that only one true label exists

(Aroyo and Welty, 2015). Precision, recall and F1 should also be corrected for chance agreement, as that varies with respect to how many labels are provided to each item. Observed precision, recall and F1 are therefore not comparable across datasets that differ in e.g. the amount of multiple labels.

When additional labels are potentially correct, intersection agreement is a good option, but only when it is corrected for chance agreement. Rather than taking the intersecting label to calculate adjusted agreement as in a soft-match measure, the data could be simulated to estimate the true chance agreement on the intersection (i.e. boot-match). This measure increases expected agreement when more labels are provided, resulting in a lower adjusted agreement. Coders should therefore only provide double labels if they are certain that both labels hold.

One limitation that needs to be taken into account when selecting the intersection label, is that the coder who provides fewer labels influences the final label more than the coder who provides more labels. Ideally, coders would therefore provide a similar proportion of double annotations. Furthermore, in an extreme case where one coder assigns an item to all categories, there would always be agreement. This is corrected for slightly in boot-match, but is more easily detected using boot-F1 and similar measures. Finally, when some labels are more frequent than others, achieving intersection agreement on this label is more likely. As a result, the intersection agreement will contain a higher proportion of dominant labels. The distribution of the intersecting labels thus does not necessarily reflect the true distribution of labels in the dataset and researchers should be careful to draw conclusions about the distributions of aggregated labels.

To conclude, if only one true label is believed to be possible for each item, augmented kappa can be used to calculate agreement in cases where annotators provide more than one label. However, if items are believed to be ambiguous, with possibly more than one true label per item, boot-match best estimates the reliability of a single final label per item. Boot-F1 and related measures reveal more about the structure of the data, such as asymmetries between the coders. Which measure is reported therefore depends on the goal of the annotation effort. Regardless, for all of these measures, chance agreement should be taken into account to make the measure comparable across datasets with different characteristics. As demonstrated above, our proposed method of bootstrapping the expected agreement can be used for this.

Acknowledgments

MM and FY are supported by the German Research Foundation (DFG) under Grant SFB 1102 (“Information Density and Linguistic Encoding", Project-ID 232722074). MS and VD are supported by the European Research Council (ERC) under the European Union’s Horizon 2020 Research and Innovation Programme, Grant 948878 (“Individualized Interaction in Discourse”).

References

Jacopo Amidei, Paul Piwek, and Alistair Willis. 2019. Agreement is overrated: A plea for correlation to assess human evaluation reliability. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 344–354.

Lora Aroyo and Chris Welty. 2013. Crowd truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard. *WebSci2013. ACM*, 2013(2013).

Lora Aroyo and Chris Welty. 2015. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Valerio Basile, Michael Fell, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, Massimo Poesio, and Alexandra Uma. 2021. We need to consider disagreement in evaluation. In *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*, pages 15–21.

Christin Beck, Hannah Booth, Mennatallah El-Assady, and Miriam Butt. 2020. Representation problems in linguistic annotations: ambiguity, variation, uncertainty, error and bias. In *14th Linguistic Annotation Workshop*, pages 60–73.

Plaban Kumar Bhowmick, Anupam Basu, and Pabitra Mitra. 2008. An agreement measure for determining inter-annotator reliability of human judgements on affective text. In *Coling 2008: Proceedings of the workshop on Human Judgements in Computational Linguistics*, pages 58–65.

Thorsten Brants. 2000. Inter-annotator agreement for a german newspaper corpus. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pages 165–172.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Ludivine Crible and Liesbeth Degand. 2019. Reliability vs. granularity in discourse annotation: What is the trade-off? *Corpus Linguistics and Linguistic Theory*, 15(1):71–99.

Anca Dumitrache, Oana Inel, Lora Aroyo, Benjamin Timmermans, and Chris Welty. 2018. Crowdtruth 2.0: Quality metrics for crowdsourcing with disagreement. *arXiv preprint arXiv:1808.06080*.

Tommaso Fornaciari, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2591–2597.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511.

Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. 2020. QADiscourse: Discourse relations as qa pairs: Representation, crowdsourcing and baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2804–2819.

Andrew Rosenberg and Ed Binkowski. 2004. Augmenting the kappa statistic to determine interannotator reliability for multiply labeled data points. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 77–80.

Ted J. M. Sanders, Wilbert P. M. S. Spooren, and Leo G. M. Noordman. 1992. Toward a taxonomy of coherence relations. *Discourse Processes*, 15(1):1–35.

Merel C. J. Scholman, Tianai Dong, Frances Yung, and Vera Demberg. 2022. Discogem: A crowdsourced corpus of genre-mixed implicit discourse relations. In *Proceedings of the Thirteenth International Conference on Language Resources and Evaluation (LREC'22)*, Marseille, France. European Language Resources Association (ELRA).

Wilbert Spooren and Liesbeth Degand. 2010. Coding coherence relations: Reliability and validity.

Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Frances Yung, Vera Demberg, and MCJ Scholman. 2019. Crowdsourcing discourse relation annotations by a two-step connective insertion task. In *Proceedings of the 13th Linguistic Annotation Workshop (LAW)*, pages 16–25.