Augmenting the kappa statistic to determine interannotator reliability for multiply labeled data points

Andrew Rosenberg
Department of Computer Science
Columbia University
amaxwell@cs.columbia.edu

Ed Binkowski
Department of Mathematics & Statistics
Hunter College
ebinkowski@juno.com

Abstract

This paper describes a method for evaluating interannotator reliability in an email corpus annotated for type (e.g., question, answer, social chat) when annotators are allowed to assign multiple labels to a message. An augmentation is proposed to Cohen's kappa statistic which permits all data to be included in the reliability measure and which further permits the identification of more or less reliably annotated data points.

1 Introduction

Reliable annotated data are necessary for a wide variety of natural language processing tasks. Machine learning algorithms commonly employed to tackle language problems from syntactic parsing to prosodic analysis and information retrieval all require annotated data for training and testing. The reliability of these computational solutions is intricately tied to the accuracy of the annotated data used in their development. Human error and subjectivity make deciding the accuracy of annotations an intractable problem. While the objective correctness of human annotations cannot be determined algorithmically, the degree to which the annotators agree in their labeling of a corpus can be quickly and simply statistically determined using Cohen's (1960) kappa measure. Because human artifacts are less likely to co-occur simultaneously in two annotators, the kappa statistic is used to measure interannotator reliability.

This paper will describe an email classification and summarization project which presented a problem for interlabeler reliability computation since annotators were allowed to label data with one or two labels (Rambow, et al., 2004). The existing kappa statistic computation does not obviously extend to accommodate the presence of a secondary label. The augmentation to the algorithm presented in this paper allows for both a more accurate assessment of interannotator reliability and a unique insight into the data and how the annotators have employed the optional second label. Section 2 will describe the categorization project. Section 3 will present a description of the annotated corpus. Section 4 will describe why the kappa statistic for determining interannotator agreement in its basic form cannot effectively be applied to this corpus. Section 5 will present a way to augment the algorithm computing kappa statistic to provide greater insight into user annotations. Section 6 will analyze the results of applying this new algorithm to the annotated corpus.

2 Project Description

This inquiry into interannotator reliability measurements was spawned by problems encountered during a project classifying and summarizing email messages. In this project email messages are classified into one of ten classes. This classification facilitates email thread reconstruction as well as summarization. Distinct email categories have distinct structural and linguistic elements and thus ought to be summarized differently. For the casual email user, the luxuries of summarization and automated classification for the dozen or so daily messages may be rather superfluous, but for those with hundreds of important emails per day, automatic summarization and categorization can provide an efficient and convenient way to both scan new messages (e.g., if the sender responds to a question, the category will be "answer", while the summary will contain the response) and retrieve old ones (e.g., "Display all scheduling emails received last week"). While the project intends to apply machine learning techniques to both facets, this paper will be focusing on the

categorization component.

3 Corpus Description

The corpus used is a collection of 380 email messages marked by two annotators with either one or two of the following labels: **question**, **answer**, **broadcast**, **attachment transmission**, **planning-meeting scheduling**, **planning scheduling**, **planning**, **action item**, **technical discussion**, and **social chat**. If two labels are used, one is designated primary and the other secondary. These ten categories were selected in order to direct the automatic summarization of email messages.

This corpus is a subset of a larger corpus of approximately 1000 messages exchanged between members of the Columbia University chapter of the Association for Computing Machinery (ACM) in 2001. The annotation of the rest of corpus is in progress.

4 Standard Kappa Shortcomings

Commonly, the kappa statistic is used to measure inter-annotator agreement. It determines how strongly two annotators agree by comparing the probability of the two agreeing by chance with the observed agreement. If the observed agreement is significantly greater than that expected by chance, then it is safe to say that the two annotators agree in their judgments. Mathematically, $K = \frac{p(A) - p(E)}{1 - p(E)}$ where K is the kappa value, *p(A)* is the probability of the actual outcome and *p(E)* is the probability of the expected outcome as predicted by chance.

When each data point in a corpus is assigned a single label, calculating *p(A)* is straightforward: simply count up the number of times the two annotators agree and divide by the total number of annotations. However, in labeling this email corpus, labelers were allowed to select either a single label or two labels designating one as primary and one as secondary.

The option of a secondary label increases the possible labeling combinations between two annotators fivefold. In the format "{<**A's labels**>, <**B's labels**>}" the possibilities are as follows: {**a**,**a**}, {**a**,**b**}, {**ab**,**a**}, {**ab**,**b**}, {**ab**,**c**}, {**ab**,**ab**}, {**ab**,**ba**}, {**ab**,**ac**}, {**ab**,**bc**}, {**ab**,**cd**}. The algorithm initially used to calculate the kappa statistic simply discarded the optional secondary label. This solution is unacceptable for two reasons. 1) It makes the reliability metric inconsistent with the annotation instructions. Why offer the option of a secondary label, if it is to be categorically ignored? 2) It discards useful information regarding partial agreement by treating situations corresponding to {**ab**,**ba**}, {**ab**,**bc**} and {**ab**, **b**} as simple disagreements.

Despite this complication, the objective in computing p(A) remains the same, count the agreements and divide by the number of annotations. But how should the partial agreement cases ({**ab**, **a**}, {**ab**, **b**}, {**ab**,**ba**}, {**ab**,**ac**}, and {**ab**,**bc**}) be counted? For example, when considering a message that clearly contained both a question and an answer, one annotator had labeled the message as primarily **question** and secondarily **answer**, with another primarily **answer** and secondarily **question**. Should such an annotation be considered an agreement, as the two concur on the content of the message? Or disagreement, as they differ in their employ of primary and secondary? To what degree do two annotators agree if one labels a message primarily **a** and secondarily **b** and the other labels it simply **a** or simply **b**? What if there is agreement on the primary label and discrepancy on the secondary? Or vice versa? In the traditional Boolean assignment, each combination would have to be counted as either agreement or disagreement. Instead, in order to compute a useful value of p(A), we propose to assign a degree of agreement to each. This is similar in concept to Krippendorff's (1980) alpha measure for multiple observers.

5 Kappa Algorithm Augmentation

To augment the computation of the kappa statistic, we consider annotations marked with primary and secondary labels not as two distinct selections, but as one divided selection.[1] When an annotator selects a single label for a message, that label-message pair is assigned a score of 1.0. When an annotator selects a primary and secondary label, a weight p is assigned to the primary label and (1-p) to the secondary label for the corresponding label-message pair. Before computing the kappa score for the corpus, a single value p where $0.5 \leq p \leq 1.0$ must be selected. If $p = 1.0$ the secondary labels are completely ignored, while if $p = 0.5$, secondary and primary labels are given equal weight. By examining the resulting kappa score at different values of p, insight into how the annotators are employing the optional secondary label can be gained. Moreover, single messages can be trivially isolated in order to reveal how each data point has been annotated with respect to primary and secondary labels. Landis and Koch (1977) present a method for calculating a weighted kappa measure. This method is useful for single annotations where the categories have an obvious relationship to each other, but does not extend to multiply labeled data points where relationships between categories are unknown.

[1] Before settling on this approach, we considered counting each annotation equivalently whether primary or secondary. This made computation of p(A) and p(E) more complex, and by ignoring the primary/secondary distinction offered less insight into the use of the labels.

5.1 Compute *p(A)*

To compute *p(A)*, the observed probability, two annotation matrices are created, one for each annotator. These annotation matrices, $M_{annotator}$, have N rows and M columns, where n is the number of messages and m is the number of labels. These annotation matrices are propagated as follows.

$M_A[x, y] = 1$, if A marked only label y for message x.

$M_A[x, y] = p$, if A marked label y as the primary label for message x.

$M_A[x, y] = 1 - p$, if A marked label y as the secondary label for message x.

$M_A[x, y] = 0$, otherwise.

Table 1 shows a sample set of annotations on 5 messages by annotator A. Table 2 shows the resulting M_A based on the annotation data in Table 1 where p=0.6.

Msg1	Msg2	Msg3	Msg4	Msg5
a,b	b,a	b	c	c,b

Table 1. Sample annotation data from labeler A

	a	b	c	d	
Msg1	0.6	0.4	0	0	
Msg2	0.4	0.6	0	0	
Msg3	0	1	0	0	
Msg4	0	0	1	0	
Msg5	0	0.4	0.6	0	
Total	1	2.4	1.6	0	5

Table 2. M_A based on Table 1 data (p=0.6;N=5).

With the two annotation matrices, M_A and M_B, an agreement matrix, *Ag*, is constructed where $Ag[x, y] = M_A[x, y] * M_B[x, y]$. A total, α, is set to the sum of all cells of *Ag*. Finally, $p(A) = \frac{\alpha}{N}$.

5.2 Compute *p(E)*

Instead of assuming an even distribution of labels, we compute *p(E)*, the expected probability, using the relative frequencies of each annotator's labeling preference. Using the above annotation matrices, relative frequency vectors, $Freq_{annotator}$, are generated. Table 3 shows $Freq_A$ based on M_A from Table 2.

$$Freq_A[y] = \sum_{x=1}^{N} \frac{M_A[x, y]}{N}$$

a	b	c	d
0.2	0.48	0.32	0

Table 3. $Freq_A$ from M_A in Table 2 (p=0.6;N=5).

Using these two frequency vectors,

$$p(E) = \sum_{y=1}^{M} Freq_A[y] * Freq_B[y].$$

5.3 Calculate *K'*

The equation for the augmented kappa statistic remains the same in the presence of this augmentation.

$$K' = \frac{p(A) - p(E)}{1 - p(E)}$$

6 Results

This technique is not meant to inflate the kappa scores, but rather to provide further insight into how the annotators are using the two labels. Execution of this augmented kappa algorithm on this corpus suggests that the annotation guidelines need revision before the superset corpus is completely annotated. (Only 150 of 380 messages present a label for use in a machine learning experiment with *K'*>0.6.) The exact nature of the adjustments is yet undetermined. However, both a strict specification of when the secondary label ought to be used, and reconsideration of the ten available labels would likely improve the annotation effort.

When we examine our labeled data, we find the average kappa statistic across the three annotators did not increase through examination of the secondary labels. If we ignore the secondary labels (p=1.0), the average *K'*=0.299. When primary and secondary labels are given equal weight (p=0.5), the average *K'*=0.281.

By examining the average kappa statistic for each message individually at different p values, messages can be quickly categorized into four classes: those that demonstrate greatest agreement at p = 1.0; those with greatest agreement at p = 0.5; those that yield a nearly constant low kappa value and those that yield a nearly constant high kappa value. These classes suggest certain characteristics about the component messages, and can be employed to improve the ongoing annotation process. Class 1) Those messages that show a constant, high kappa score are those that are consistently categorized with a single label. (92/380 messages.) Class 2) Those messages with a constant, low kappa are those messages that are least consistently annotated regardless of whether a secondary label is used or not. (183/380 messages.) Class 3) Messages that show greater agreement at p = 1.0 than at p = 0.5 demonstrate greater inconsistency when the annotators opt to use the secondary labels but are in (greater) agreement regarding the primary label. Whether the primary label is more general or more specific depends on, hopefully, annotation standards, but in the absence of rigorous instructions,

individual annotator preference. (58/380 messages.) Class 4) Messages that show greater agreement at p = 0.5 than at p = 1.0 are those messages where the primary and secondary labels are switched by some annotators, the above {**ab**,**ba**} case. From inspection, this most often occurs when the two features are not in a general/specific relationship (e.g., **planning** and **question** being selected for a message that contains a question about planning), but are rather concurrent features (e.g., **question** and **answer** being labeled on a message that obviously includes both a question and an answer). (47/380 messages.) Each of the four categories of messages can be utilized to a distinct end towards improvement of annotation instructions and/or annotation standards. Class 1 messages are clear examples of the labels. Class 2 messages are problematic. These messages can be used to redirect the annotators, revise the annotation manual or reconsider the annotation standards. Class 3 messages are those in which annotators use the optional secondary label, but not consistently. These messages can be employed to reinstruct the annotators as to the expected use of the secondary label. Class 4 messages pose a real dilemma. When these messages in fact do contain two concurrent features, they are not going to be good examples for machine learning experiments. While representative of both categories, they will (most likely) at feature analysis (the critical component of machine learning algorithms) be poor exemplars of each. While the fate of Class 4 messages is uncertain[2], identification of these awkward examples is an important first step in handling their automatic classification.

7 Conclusion

Calculating a useful metric for interannotator reliability when each data point is marked with optionally one or two labels proved to be a complicated task. Multiple labels raise the possibility of partial agreement between two annotators. In order to compute the observed probability (p(A)) component of the kappa statistic a constant weight, p, between 0.5 and 1.0 is selected. Each singleton annotation is then assigned a weight of 1, while the primary label of a doubleton annotation is assigned a weight of p, the secondary 1-p. These weights are then used to determine the partial agreement in the calculation of p(A). This augmentation to the algorithm for computing kappa is not meant to inflate the reliability metric, but rather to allow for a more thorough view of annotated data. By examining how the annotated components of a corpus demonstrate agreement at varying levels of p, insight is gained into how the annotators are viewing these data and how they employ the optional secondary label.

8 Future Work

The problem that spawned this study has led to further discussions about how to get the most information out of apparently unreliably labeled data. The above process shows how it is possible to classify messages into a few categories by their reliability at different levels of p. However, even when interlabeler reliability is relatively low, annotated data can be leveraged to improve the confidence in assigning labels to messages. Annotators can be ranked by "how well they agree with the group" using kappa. Messages (or other labeled data) can be ranked by "how well the group agrees on its label" using variance or $-p*\ln(p)$. Annotator rankings can be used to weight "better" annotators greater than "worse" annotators. Similarly, message rankings can be used to weight "better" messages greater than "worse" messages. The weighted annotator data can be used to recompute the message weights. These new message weights can then be used to recompute annotator weights. Repeating this alternation until the weights show minimal change will minimize the contributions of unreliable annotators and poorly annotated messages to the assignment of labels to messages, thereby increasing confidence in the results. An implementation of this "sharpening" algorithm is currently under development.

Acknowledgments

Thanks to Becky Passonneau for her insightful comments on an intermediate draft. This work would not have been possible without the support and advice of Julia Hirschberg, Owen Rambow and Lokesh Shrestha. This research was supported by a grant from NSF/KDD #IIS-98-17434.

References

J. A. Cohen. 1960. *Educational and Psychological Measurement*, 20(1):37-46.

Klaus Krippendorff. 1980. *Content Analysis, an Introduction to Its Methodology*. Thousand Oaks, CA.

J. R. Landis and G.G. Koch. 1977. *Biometrics*, 33(1):159-174

Owen Rambow and Lokesh Shrestha and John Chen and Charles Lewis. 2004. *Summarizing Email Threads*. Under submission.

[2] One potential solution would be to create a new annotation category for each commonly occurring pair. While each Class 4 message would remain a poor exemplar of each component category, it would be a good exemplar of this new "mixed" type.