Analyzing LEOKA Narratives with Machine Learning: Codebook

Javier Osorio

University of Arizona

Version 1.2

March 2024

Contents

1 INTRODUCTION **2**

2 TEAMS OF CODERS **3**

3 CODING PROTOCOL **4**
- 3.1 Classification . . . 4
- 3.2 Annotation . . . 6

4 CODING INTERFACE **8**

5 CODING PROCEDURE **10**
- 5.1 High-Quality Annotations . . . 12
- 5.2 Special Classification Cases . . . 16
- 5.3 Special Annotation Cases . . . 17

6 ADDITIONAL FEATURES **19**

7 IMPROVING THE CODING QUALITY **21**
- 7.1 Preliminary Steps . . . 24
- 7.2 Review Process . . . 25

8 GRADING **27**

1 INTRODUCTION

In this project, we will be coding narratives of Law Enforcement Officers Killed and Assaulted (LEOKA) generated by the Federal Bureau of Investigation (FBI). These annotations will then be used as input for Machine Learning (ML) with the purpose of training different algorithms to classify and extract information at a large scale.

Since 1973, the FBI uses the Uniform Crime Reporting (UCR) to publish an annual Law Enforcement Officers Killed and Assaulted report providing information about officers who were killed (feloniously or accidentally) or assaulted while on duty. More information about LEOKA is available at https://ucr.fbi.gov/leoka/2019/resource-pages/about-leoka.

The narratives used in this study were collected directly from the FBI LEOKA reports over the years. They include public accounts of killings and assault incidents of law enforcement officers across multiple US states between 1996 and 2019.

The purpose of this activity is to expose undergraduate students to the use of computational social sciences with direct applications to the security and criminal justice sector. Most students in social sciences could benefit from gaining experience on how to conduct annotations for ML, and use ML and data science techniques to address important problems in crime, security, and public policy. With the prevalent use of Artificial Intelligence in every aspect of our personal and professional life, it is crucial for students to gain familiarity with these tools and methodologies as they become part of the work force. The more familiarity and skills students have with the use of data and technology focused on social science applications, the better prepared they will be to meet the challenges of a successful professional career.

2 TEAMS OF CODERS

For this activity, we will be dividing the class into teams of LEOKA coders. Each team will include 3-4 students who will be in charge of coding different subsets of LEOKA narratives. It is important to have multiple coders coding the same stories in order to generate a sense of agreement about their coding. The level of agreement is technically known as inter-coder reliability, and it serves as a key measure of the quality of the information that we provide to the ML algorithms.

ML algorithms learn based on a lot of examples. When we expose a ML algorithm to thousands of examples, the machine eventually identifies patterns that allow it to understand the underlying logic behind the decision rules operating behind those examples. In order to train the ML algorithms to make accurate decisions, we need to provide high-quality examples. Otherwise, the ML algorithm will be learning from confusing or inconsistent decisions, and will likely perform poorly when coding data on its own. The logic is very simple, "garbage in, garbage out.". If we provide annotations full of inconsistencies and coding errors, then the ML algorithm will learn from such low-quality set of examples; the result will be also full of errors.

Students will be able to identify the other members of their LEOKA team in D2L under the "LEOKA Coding" assignment. There you will find the name and email address of your team members. As you work on this assignment, please stay in touch with the other members of your LEOKA team, also enable a quick and effective communication channel with them (e.g. chat group), so that you can quickly bring up questions and solve them together. Students are also strongly encouraged to meet (in person or online) to have coding sessions together.

3 CODING PROTOCOL

For this project, we will be conducting two types of tasks:

- **Classification:** coders will classify the narrative according to several characteristics. Classification takes place at the document level and is implemented by selecting categories from a drop-down measure.
- **Annotation:** coders identify specific words in the text of the narrative to identify certain relevant elements or details of the event. Annotations take place at the word or phrase level.

3.1 Classification

The classification provides a general description of some key characteristics of the incident. Classification occurs at the narrative (or document) level. In this project, coders will classify the LEOKA narratives according to five main categories: killed, assaulted, assignment, situation, and suspect. These main categories are disaggregated in the following sub-categories:

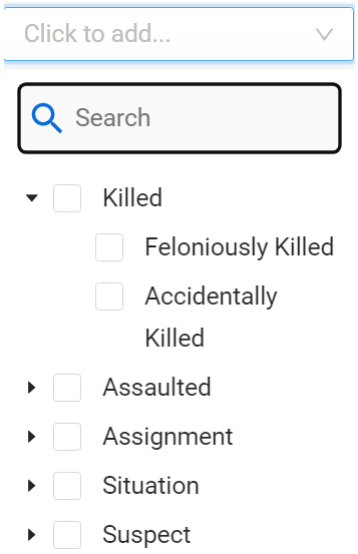
- Killed: Indicates whether the law enforcement officer was killed or not. It also identifies if it was an intentional or accidental event.
  - Feloniously Killed
  - Accidentally Killed
- Assaulted: Indicates whether the officer was assaulted or not. In addition, it indicates if the officer was injured or not.
  - Not injured
  - Injured
- Assignment: Describes the type of assignment or activity that led the officer to participate in the situation that led to the event. This includes the following subcategories:
  - Conducting arrest

  - Citizen call
  - Drug warrant
  - Investigation
  - Police call
  - Traffic stop
- Situation: Describes the type of situation in which the officer and the perpetrator interacted. The type of engagement is disaggregated in the following sub-categories:
  - Ambush
  - Encounter
  - Pursuit
- Suspect: Indicates the result of the event for the perpetrator of the killing or assault. This characteristic includes the following sub-categories:
  - Arrested
  - Escaped
  - Injured
  - Killed
  - Suicide

When classifying the LEOKA narratives in the annotation interface, coders will be able to identify the different categories using the drop-down menu from the coding interface as displayed in Figure 1. Each category in the drop-down menu unfolds its corresponding sub-categories.

Coders should focus on classifying the **specific sub-category** that corresponds to the description of the narrative, rather than the higher-order category. Coders should try to identify **all** the five different sub-categories that describe the content of the event narrative. Please **do not select** the main generic category (e.g. Killed, Assaulted, Assignment, Situation, Suspect); instead, please classify the subcategory indicating the detailed characteristics

Figure 1: Classification Drop-down Menu



of the incident.

Please note that you do not need to annotate the text of the narrative to classify the type of event described in the document. All you have to do is classify the event from the drop-down menu in the interface.

3.2 Annotation

After classifying the event categories, coders should annotate specific elements that provide more detail about the incident. The annotation includes the following categories:

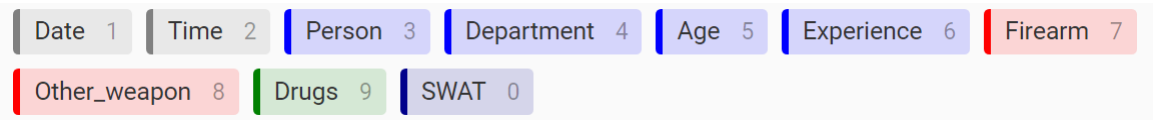
- DATE: This is self-evident.
- TIME: Same here.
- PERSON: Do not expect proper names here. This is usually a generic description of the type of officer (e.g. sheriff, agent, police officer).
  - Please focus exclusively on annotating information about law enforcement officers killed or attacked. Do not annotate information about suspects. Do not annotate information about supporting officers not involved in the incident.
- DEPARTMENT: LEOKA narratives generally indicate the police department of the office

involved in the incident.

- AGE: This is the age of the law enforcement officer.
  - Do not annotate information about suspects.
- EXPERIENCE: Indicates the officer's years of experience if mentioned in the narrative.
  - Please highlight the whole phrase e.g. "`15 years of experience in law enforcement`" rather than just "`15 years`".
- Firearm: Indicates the type and caliber of firearm mentioned in the narrative, if there is one involved in the incident.
- OTHER WEAPON: Indicates if there was any other type of weapon other than a firearm involved in the incident.
- DRUGS: Indicates the type of drugs, if any, that are mentioned in the event narrative.
- SWAT: Indicates if the incident involved the deployment of a SWAT team.

Figure 2 shows the way the coding interface displays the annotation categories. Notice that they are grouped by color with gray denoting DATE and TIME; blue denotes officer characteristics including PERSON, DEPARTMENT, AGE, and EXPERIENCE; red indicates the type of weapon used in the incident such as FIREARM and/or OTHER WEAPON; green indicates the presence of DRUGS in the incident; and dark blue indicates the deployment of SWAT team in the event. To conduct the annotations in the coding interface, coders need to select specific words or segments in the text of the LEOKA narrative.

Figure 2: Annotation Menu



4 CODING INTERFACE

To process these LEOKA narratives, we will be using LABEL STUDIO, a web-based annotation interface. All students must have received an email invitation from LABEL STUDIO asking them to join this annotation space. Please check your inbox and spam folders. We will be using an academic license that only works for your University of Arizona email address. So, make sure you use your `netid@arizona.edu` credentials when joining LABEL STUDIO.

When opening LABEL STUDIO, coders will see the Data Manager screen as displayed in Figure 3. The Data Manager contains the list of narratives assigned to each team of coders. In total, each coder will process **130 narratives** randomly selected from a large collection of event descriptions generated by the FBI. These narratives are usually a short paragraph describing the events unfolded in the incident, although sometimes the description can be a little longer. As discussed above, members of the same team will be coding the same narratives to create inter-coder reliability. The first 30 narratives in each set are common to all coders. This will help us to generate an overall inter-coder reliability assessment. The rest 100 narratives are exclusive to the coders in each LEOKA team. To start coding, students should click on the "**Label All Tasks**" button.

Figure 3: Data Manager

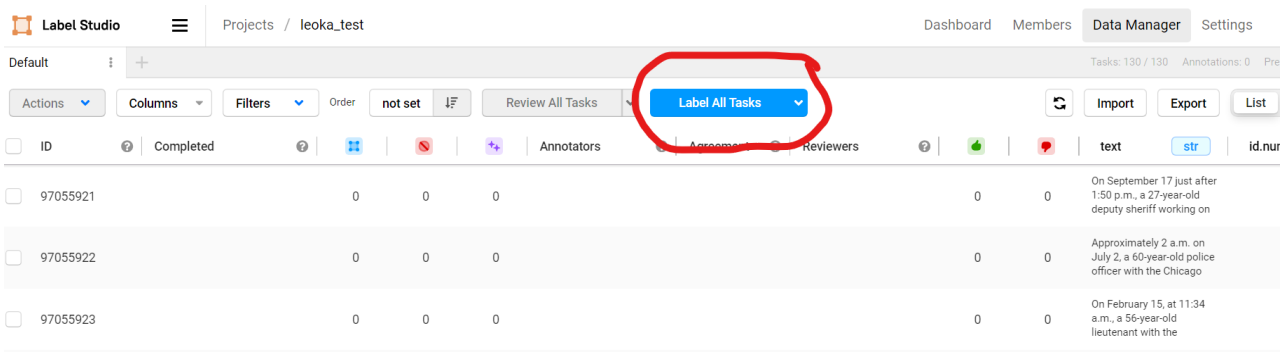
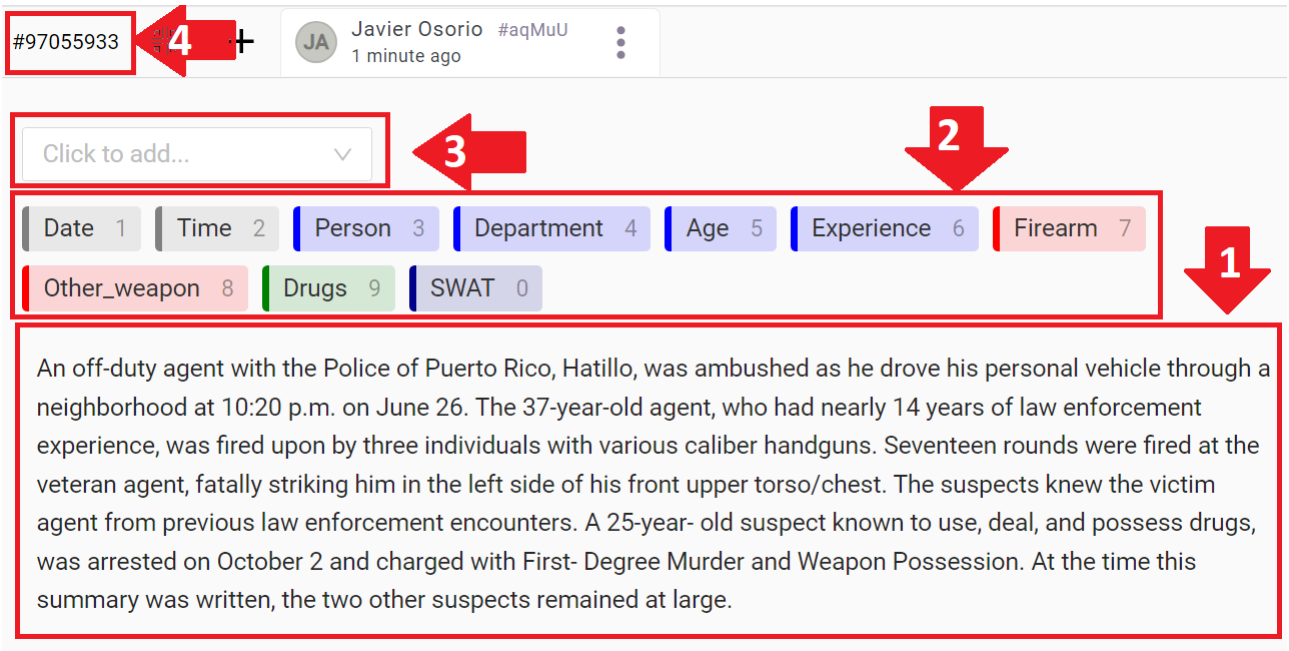


Figure 4 displays the coding interface. The interface presents the text of the LEOKA narrative as indicated in frame 1. The menu of annotations displays the specific categories to be highlighted in the text as denoted in frame 2. To unfold the drop-down menu of event

categories, click on the area highlighted in frame number 3. Finally, to keep track of the specific narratives, frame 4 in the upper-left corner indicates the narrative ID.

Figure 4: Coding Interface

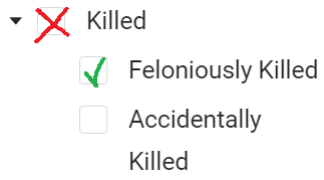


5 CODING PROCEDURE

Before coding the LEOKA narratives, please read the codebook a couple of times paying particular attention to the classification and annotation categories. To conduct the annotations, please implement the following steps.

- **Step 1 - Read**: Read the narrative of the event without making any annotation or classification.
- **Step 2 - Annotate**: As you read the narrative a second time, please annotate in the text of the narrative **all** the relevant aspects of the event including the date, time, person, police department, age of the officer, years of experience, firearm or other weapons used, indications of drugs, and SWAT team deployment as they are mentioned in the narrative.
  - To annotate any of these features, first click on the Annotation Menu the specific characteristic you want to annotate.
  - Then highlight the text of the narrative the word or span of words that contain the element you want to annotate.
  - When highlighting the text, be careful to:
    * Select only the relevant words.
    * Avoid including in your selection the blank space before or after a relevant word. For example, instead of selecting "` pistol `", simply select "`pistol`" without the blank spaces before and after the word.
    * Avoid selecting articles or prepositions when annotating words. For example, instead of selecting "`the police officer`", exclude the article "the" and simply select "`police officer`".
- **Step 3 - Classify**: After reading the narrative, click on the drop-down menu to classify the key characteristics that describe the incident.
  - Classifying the **specific sub-category** that corresponds to the description of the narrative, rather than the higher-order category. Please **do not select** the

main generic category (e.g. Killed, Assaulted, Assignment, Situation, Suspect); instead, please classify the subcategory indicating the detailed characteristics of the incident. For example, instead of selecting "Killed" as the main category, you should select the specific sub-category "Feloniously Killed" as indicated below:



  - Coders should try to identify **all** the five different sub-categories that describe the content of the event narrative if they are mentioned in the narrative.

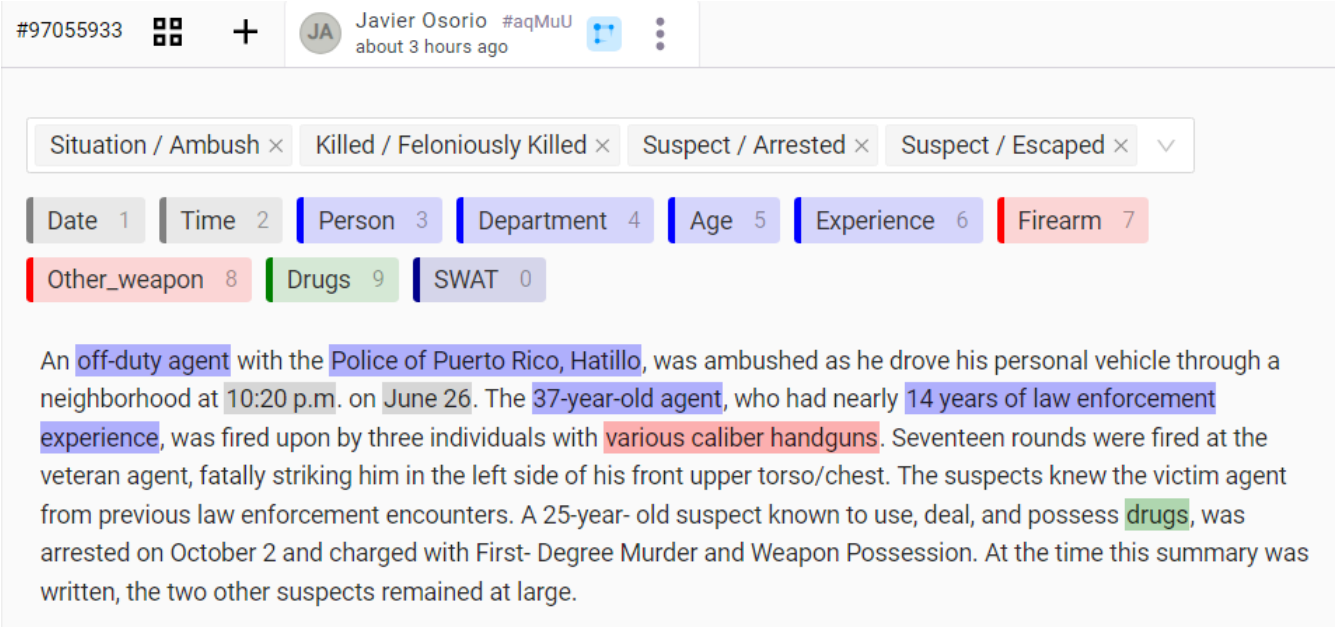
- **Step 4 - Verify**: Please verify your annotations and classifications.
  - Make sure all the relevant characteristics are annotated in the narrative.
    * A simple way to check the annotations, is to simply click on each of the tabs in the annotation menu (see Figure 2) and the selected text will highlight. If any of the tabs do not highlight, that should reveal if you have missing annotation categories.
  - Verify that the classification includes all relevant characteristics.
    * Please make sure you select all relevant sub-categories in the drop-down menu.
    * Also, make sure there are no main categories (e.g. Killed, Assaulted, Assignment, Situation, Suspect) accidentally selected.
- **Step 5 - Submit**: To complete your coding task, click on the **Submit** button at the bottom.

Figure 5 shows an example of a completely coded event. This narrative classifies the incident with the following characteristics:

- SITUATION: Ambush
- KILLED: Feloniously Killed
- SUSPECT: Arrested

- SUSPECT: Escaped
- DATE: "June 26"
- TIME: "10:20 p.m."
- PERSON: "off-duty agent"
- DEPARTMENT: "Police of Puerto Rico, Hatillo"
- AGE: "37-year-old agent"
- EXPERIENCE: "14 years of law enforcement experience"
- FIREARM: "various caliber handguns"
- DRUGS: "drugs"

Figure 5: Example of Coded Narrative



5.1 High-Quality Annotations

Please always keep in mind that **quality is not an accident**. Please, pay attention and care in your annotation task. Try to devote sufficient time to conduct your annotations to make sure you devote sufficient attention and care to what you do. After all, what you do is a reflection of who you are. So, I would like to see the best of you reflected in the work

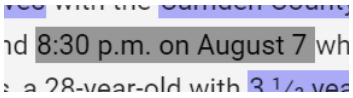
that you produce. Here are a few simple but important tips on how to avoid low-quality classification and annotations.

Highlight your words carefully. When conducting annotations, please be careful in the way you select the relevant words. Avoid sloppy selections that only capture a fraction of the word. Instead, be intentional and careful in selecting the relevant words. Here are a few examples of improper/proper selections in blue:

- Instead of selecting "`off-dutty officer`" in a sloppy and incomplete way, you should carefully select the words "`off-duty officer`".
- Instead of selecting "`use, deal, and possess drugs`", you should select only the word "`drugs`". The other parts of the sentence are not very relevant to identify the mentions of drugs.
- Instead of selecting "`14 years of law enforcement experience`", you should select "`14 years of law enforcement experience`". This helps to reduce ambiguity around the years of experience as opposed to any other kind of years.

Do not conflate date and time. Please do not conflate the date and time in the same selection. Figure 6 shows an example of an erroneous annotation conflating the date and time inside the same selection. Instead, these should be annotated separately as "`8:30 p.m.`" for time and "`August 7`" for date.

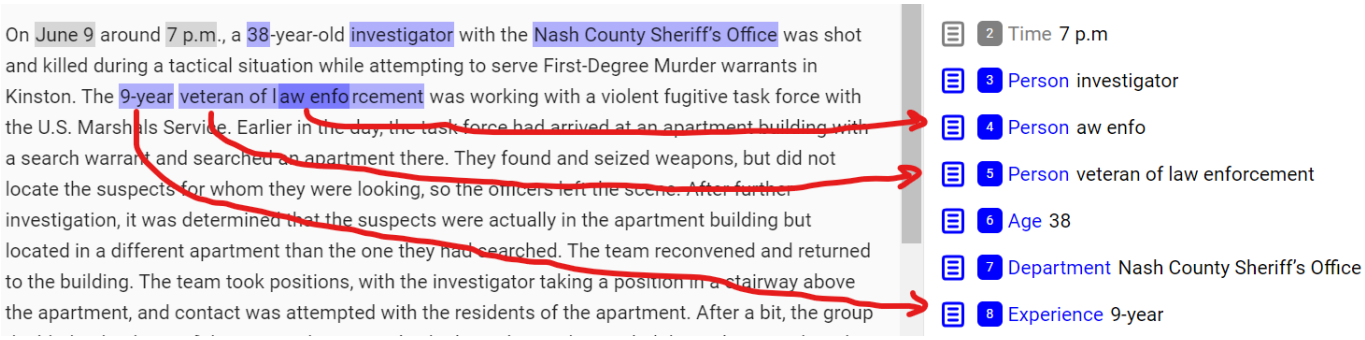
Figure 6: Example of Conflated Date and Time



Avoid chopped selections. To conduct your annotations, please select the whole sentence. The example below in Figure 7 is particularly bad. The right way of annotating the years of experience is by selecting "`9-year veteran of law enforcement`". However, the annotation below was so poorly done that it generated three separate annotations, one indicating "`9-year`" but without providing additional context, another one indicating "`veteran of law enforcement`" without the years, and the last one makes absolutely no sense "`aw`

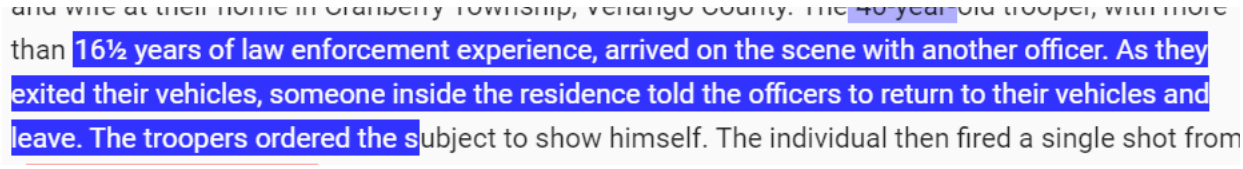
`enfo`". Please avoid this type of sloppy annotation. They just generate garbage that will confuse the ML algorithm.

Figure 7: Example of Sloppy Annotation



Select only the relevant text. When conducting your annotations, please select only the relevant text. Figure 8 below shows a remarkably long annotation for years of experience. Instead of just annotating "`16`$\frac{1}{2}$ `years of law enforcement`", the coder annotated more than two rows of irrelevant text.

Figure 8: Example of Wrong Long Annotation

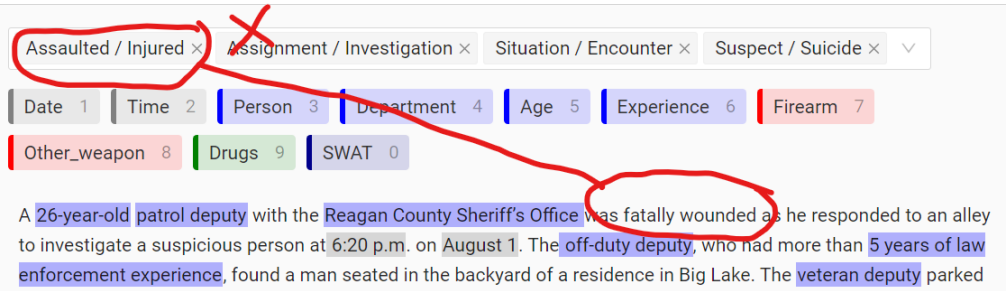


Consider Killed and Assaulted as mutually exclusive for the same officer. The categories of "Killed" and "Assaulted" should be considered mutually exclusive when referring to the same officer. Many narratives provide information about an officer getting shot and later dying. Please code the incident as "Killed/Feloniously Killed" if the officer unfortunately dies from the wound. Since the killing is a more severe type of incident than an assault, then the "Assaulted/Injured" becomes irrelevant and should not be coded. So, if the narrative indicates first an attack against an officer who ends up dying, let's classify only the killing and not both the killing and the assault. **Attention:** if the incident talks about multiple officers, one getting killed and another one getting injured, then we should classify

the first officer as "Killed/Feloniously Killed" and the second one as "Assaulted/Injured". In such case, the "Killed" and "Assaulted" are not mutually exclusive because they refer to different officers.

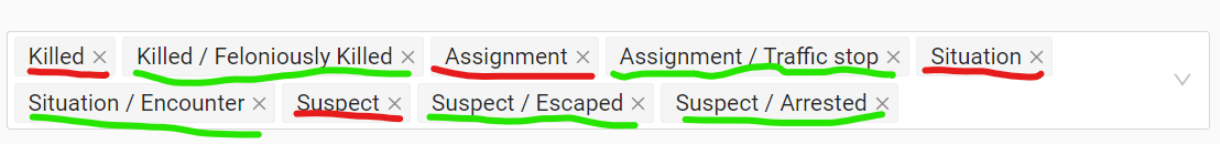
Fatally Wounded. Coders may encounter narratives indicating that an officer was fatally wounded. The indication of "fatally" means that the police officer unfortunately died because of the injury. In those cases, please classify the incident as "Killed/Feloniously Killed". As you can see in Figure 9, an incident of a "fatally wounded" officer was erroneously classified as "Assaulted/Injured".

Figure 9: Example of Wrongly Classified Assaulted/Injured



Select the Proper Categories. When classifying categories, make sure to select the **relevant sub-categories** and not the main categories (e.g. Kill or Suspect) without specifying what happened. Figure 10 below shows in red the categories wrongly selected and in green the correct category selection. Notice that the categories underlined in red only indicate the main category but provide no detail about the incident. In contrast, the categories underlined in green are correctly selected as they indicate the specifics of the event.

Figure 10: Example of Wrong and Correct Category Selection



Complete All Aspects of the Coding Schema. When classifying and annotating the information, make sure you:

- Complete the classification of all categories.

  - The review shows many stories with missing categories related to "Assignment" and "Situation". Please make sure you mark all the relevant dimensions of the incident.
  - Please make sure you mark all the relevant aspects of the story before moving on.
- Make sure you effectively annotate all relevant characteristics of the event.
  - The review shows several stories with missing information such as "Person" and "Department", but there could be other missing characteristics. Please just make sure you cover all the different aspects of the incident.
  - Please make sure you mark all the relevant annotations before moving on.

5.2 Special Classification Cases

1. **Situation Not Listed in the Drop-Down Classification Menu**. If you encounter a situation not listed in the drop-down menu, you do not need to classify that aspect of the narrative. For example, if the situation refers to agents serving an arrest warrant, officers transporting a prisoner, etc. Those types of situations are not listed in the drop-down menu. So, we do not need to classify those aspects of the narrative.
2. **Officers injuring themselves by accident**. If the narrative indicates that a law enforcement injured himself by accident (e.g. stepping on a nail, cutting himself with glass, hurling an ankle when jumping a fence), then do not classify that as an assault. We only want to code actions of killing or assault perpetrated by the suspect.
3. **Non-law enforcement officers getting killed or injured**. Sometimes, LEOKA narratives provide information about other people getting killed or injured other than the law enforcement officers. For example, the narrative could provide information about firefighters being attacked, a bystander getting killed, a store manager getting injured, a neighbor getting shot, etc. In those cases, please do not classify or annotate information about people injured or killed if they are not law enforcement officers. Remember, the scope of this research is to code only information about law enforcement

agents killed or assaulted.

5.3 Special Annotation Cases

1. **Repeated Information**. LEOKA narratives are sometimes redundant repeating some aspects of the information several times. For example, there could be the case when the story initially mentions a "43-year-old police officer", then keeps referring to that same "43-year-old police officer" in a subsequent sentence and mentions the "43-year-old police officer" again toward the end of the narrative. In those cases, please only code the first appearance of the relevant information and not the subsequent ones. That type of redundancy does not bring new information to the ML learning process and increases the chance of coding errors (with some coders missing those redundant annotations) that could undermine the inter-coder reliability.
2. **Suspect Information**. We do not need to annotate information about the suspect (e.g. age). The blue annotation tabs are only for law enforcement officers. So, please do not use the blue tabs to annotate information about the suspects. We only classify with the drop-down menu what happens to the subject at the end of the story (e.g. arrested, escaped).
3. **Multiple Officers**. If you encounter a narrative where there are multiple police officers, please annotate the characteristics of each of them (e.g. person, age, department, etc.) as long as something happened to them (e.g. killed or injured).

   If additional police officers show up to the scene (e.g. as backup) but they are just part of the context and were not killed or injured, then we do not need to code information about them. Remember, we are coding incidents of assault or killings of police officers. So, we should narrow our scope to focus on officers who get injured or die.
4. **Years of Experience**. When coding the officer's years of experience, please select the whole sentence. For example, instead of selecting "`12 years`", please select "`12 years of law enforcement experience`". This context will help to disambiguate the

information in the ML process.

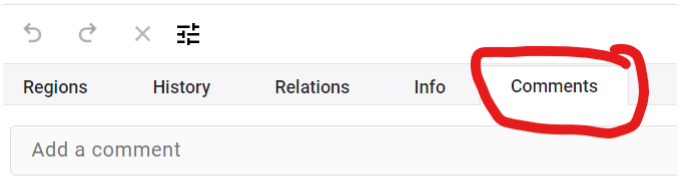
5. **Firearms**. When coding incidents involving firearms, please focus on the firearm used by the suspect. Please do not code the firearm the law enforcement officer used (e.g. service weapon) to shoot at the suspect, unless the suspect grabbed it to shoot at the officer with his own weapon.
6. **Weapons not used**. If the narrative mentioned that police seized a weapon (e.g. maybe in a house search or traffic stop), but that weapon was not used to attack or kill the law enforcement officer, then please do not annotate the weapon. We want to train the ML algorithm to identify the weapons used to attack officers, not just to detect a weapon just mentioned tangentially.
7. **Vehicles intentionally used to attack or kill**. If the incident involved a suspect intentionally using a vehicle to run over a law enforcement officer, then please annotate the vehicle as "other weapon". This annotation is only valid when the vehicle was used with the deliberate intention to target the officer. If the narrative indicates that the officer was killed or injured in a traffic accident (e.g. car crash or truck running over an officer), then do not code the vehicle as a weapon.

6 ADDITIONAL FEATURES

The LABEL STUDIO interface has a few additional features that you may find useful.

Comments. If you have questions or comments, you can leave some notes to yourself by clicking on the `Comments` tab as displayed in Figure 11.

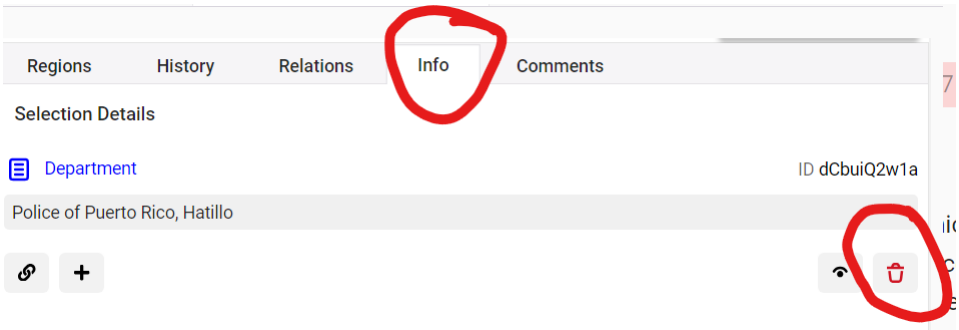
Figure 11: Insert Comment



Delete an annotation. If you made an annotation mistake and want to edit or delete an annotation, follow this procedure as indicated in Figure 12:

- In the text of the narrative, click on the annotation you want to edit or delete.
- click on the `Info` tab.
- Verify the text of the annotation to make sure this is the one you want to edit or delete. For example, here we will delete the DEPARTMENT annotation "Police of Puerto Rico, Hatillo".
- Click on the trash can icon to delete the annotation.
- Create a new annotation in the text.

Figure 12: Delete Annotation



Demonstration Video. Please click on the link below to see a brief video of the coding process in Label Studio. Please excuse my terrible voice.

https://tinyurl.com/leokavideo1

7 IMPROVING THE CODING QUALITY

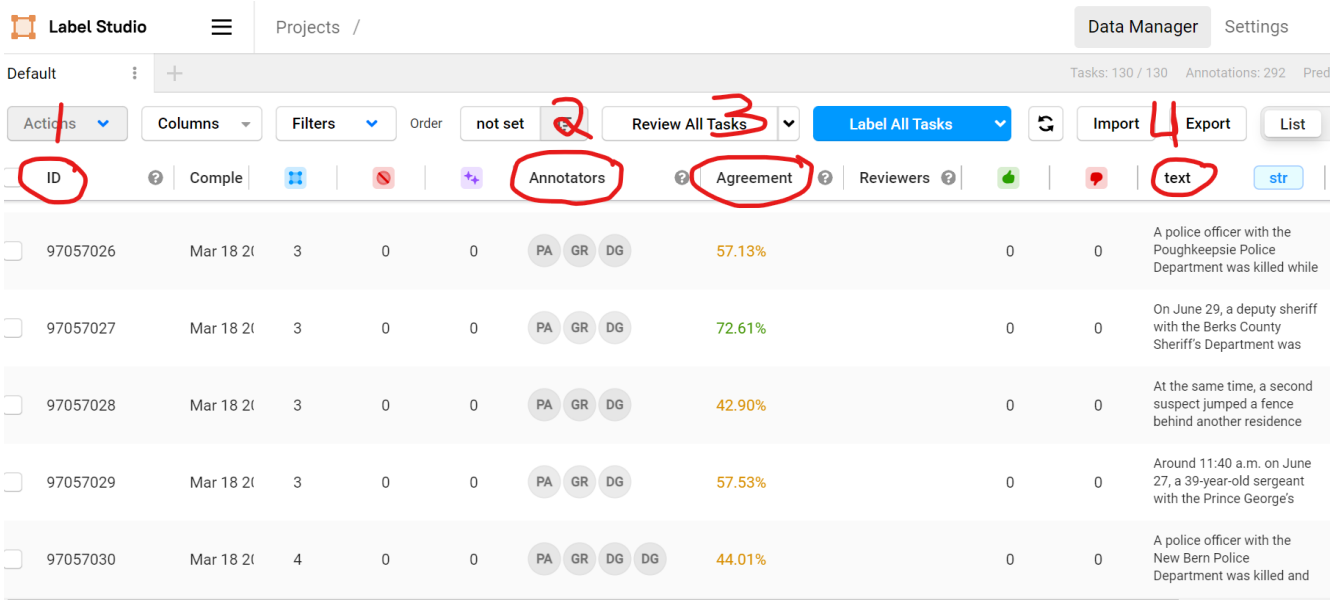
The first task of this coding exercise is to complete processing all the LEOKA narratives assigned to each team. However, completing the coding exercise does not mean that the classifications and annotations were properly done. For this reason, the second task focuses on reviewing and improving the quality of your coding. Remember, the key reason for providing high-quality annotations is because Machine Learning algorithms learn based on examples. If we provide low-quality annotations, filled with errors, confusing decisions, chopped sentences, etc., then the ML algorithm will learn from low-quality examples and perform poorly. In contrast, if we provide high-quality annotations, the ML algorithm will have solid and reliable training data to learn from, and the result should be high-quality performance.

To improve the quality of the annotations, coders will be upgraded from "Annotator" to "Manager" in Label Studio. This upgraded user status will enable you to edit your annotations while seeing how other students coded their narratives. At this point, we do not have a "Gold Standard Record" or "Ground Truth" against which to compare our annotations. So, we are going to use the comparison method to contrast the coding decisions made by multiple users to increase the level of agreement between coders.

Figure 13 below shows the Data Manager dashboard. There you will be able to see (1) the ID of each specific LEOKA narrative, (2) the coders who contributed to coding each narrative, (3) the degree of agreement between coders in each story, and (4) the text of each narrative. To keep your working space ordered and systematic, I recommend sorting all your narratives by ID. You can do that by clicking on the "Order" box in the top menu, then select the variable "ID", and then clicking the arrow down to order in ascending order. Always verify the ID number sequence to ensure you sorted the narratives in the desired order. As you review your annotations, I strongly recommend keeping a notebook handy to track the ID number of the latest narrative reviewed. In that way, you can easily pick up the review process exactly where you left it.

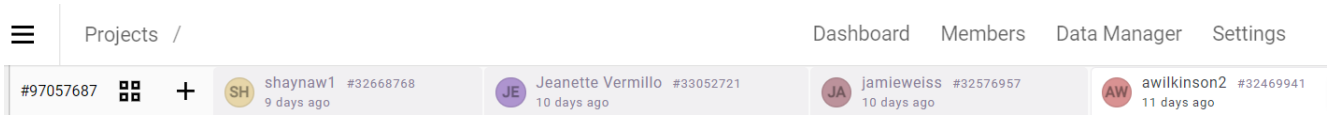
To start reviewing your annotations, click on the LEOKA narratives, I strongly recom-

Figure 13: Data Manager View for Managers



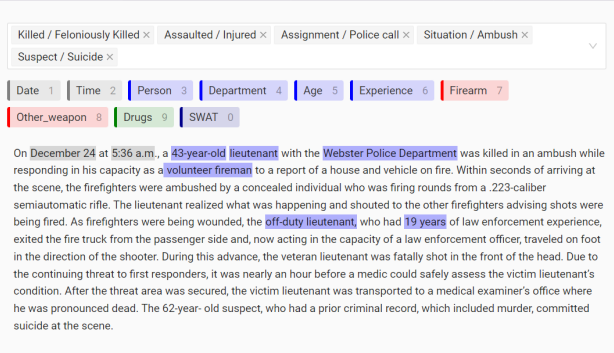
mend sorting them first to start from the top and then move progressively downwards. Once you click on a narrative, you will be able to see the annotations on the screen. On top of the annotation interface, the Coders Menu shows tabs for the different members of your team as illustrated in Figure 14. Please identify your own Coder Tab to display your annotations.

Figure 14: Coder Menu

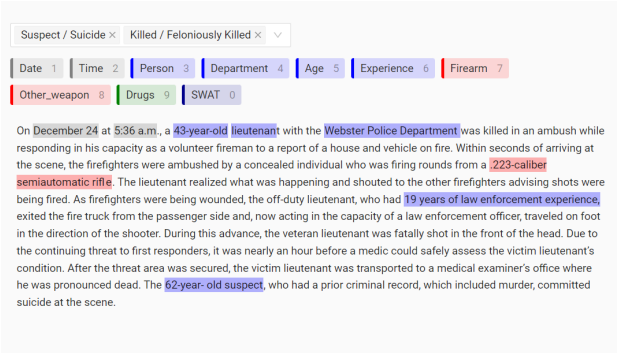


To improve the quality of your coding and increase your level of agreement with other coders, please compare your classifications and annotations to the other coders. Figure 15 below shows an example of how different coders make different decisions about the categorization and text annotations done on the same annotation. In this example, there are variations in the number and type of categories characterizing the incident as well as discrepancies in the annotations highlighted in the text. Some of those differences are due to inadequate application of the coding rules discussed in the sections above or lack of attention.

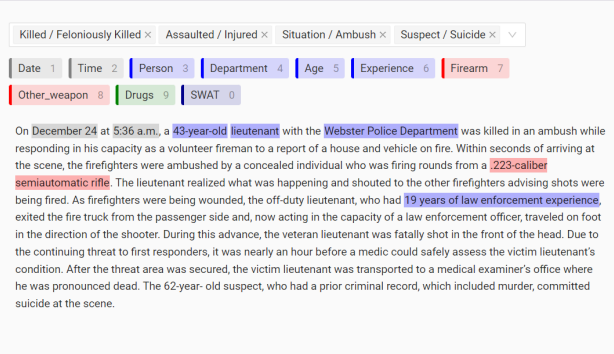
Figure 15: Example of Same Text Coded by Multiple Coders



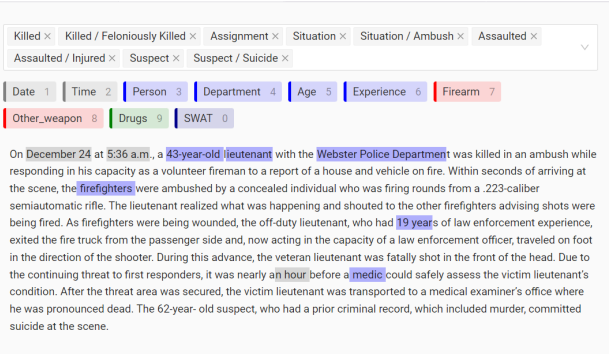
(a) Coder 1



(b) Coder 2



(c) Coder 3



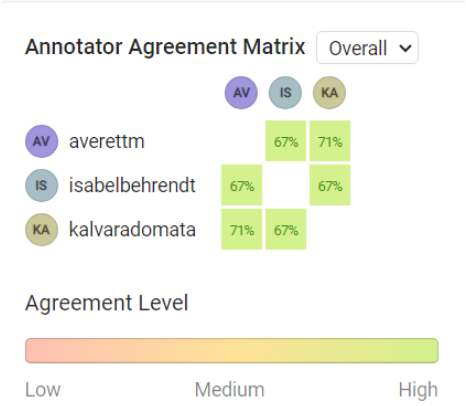
(d) Coder 4

To improve the quality of your coding, please compare your classification and annotations to those of your peers. By comparing the different coding decisions, you will identify the classifications or annotations that you got right, and those where you could improve the quality of your coding. By contrasting your coding to that of your peers, you may realize that you missed some elements in the annotations, or you misclassified the incident to a type of event that does not correspond with the narrative details, or the text that you highlighted did not include the right or complete elements of the text.

As Figure 16 shows, the Annotation Agreement Matrix in Label Studio already generates an inter-coder reliability score indicating the degree of agreement between coders. The system shows the inter-coder reliability between pairs of coders in a matrix form. This helps to identify the pairs of coders with the highest level of agreement and the ones with the lowest performance. As you make progress on reviewing your annotations and improving the quality

of your coding decisions, the inter-coder reliability will increase. In general, it is necessary to reach at least a 75% agreement between human coders to consider the annotation valid.

Figure 16: Example of Inter-coder Reliability Score

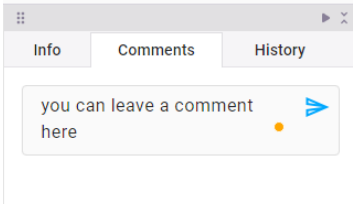


7.1 Preliminary Steps

Before reviewing and editing your coding decisions, I strongly encourage coders to take the following preparatory steps:

1. Read again the "Coding Protocol" in section 3.
2. Read in detail the "Coding Procedure" in section 5.
3. Review the editing features as indicated in the "Additional Features" in section 6. This will remind you about the procedure to delete and edit annotations that need improvement.
4. Keep the codebook handy to use as a reference when processing the narratives.
5. Please use the "Comments" tab to give feedback to your peers (Figure 17). This will help you communicate with your team members to let them know when there is an issue in one of their classification or annotation decisions.
6. When providing comments to your peers, always be respectful, indicate the name of the coder you are referring to, and clearly articulate the issue that called your attention. This will help the other coders and yourself to quickly navigate the list of annotations and address your comment.

Figure 17: Comments Tab



7. After addressing a comment and improving your coding, please mark the comment as "Resolved". This will help you to clear your annotation space and focus on the pending issues that need attention.

7.2 Review Process

The overall objective of the review process is to improve the quality of collective effort by increasing our levels of inter-coder reliability. The more agreement we have among coders, we can be more confident about the quality and consistency of the classification and annotation decisions in the data. To review and improve the quality of your classification and annotation decisions on the LEOKA narratives, please follow these steps:

- **Step 1. Sort the Narratives by ID**. Please log in to LABEL STUDIO and go to the Data Manager page. On the top menu, you will be able to see the "Order" function as displayed in Figure 18. Sort your stories by ID in descending order. This will help you conduct your review in an ordered and progressive manner. When you pause your work session, remember to **keep track of the latest ID** that you worked on. This will allow you to resume your annotations right where you left them.

Figure 18: Sorting narratives

- **Step 2. Keep the codebook handy**. The codebook is a resource for you. Please use it.

- **Step 3. Review your annotations**. First focus on reviewing the annotations highlighted in the LEOKA narratives.
  - First focus on your annotations.
  - As you read the sentence, click on the highlighted text to make sure you annotated those text segments according to the right annotation category.
  - Make sure you are selecting the words with precision. Avoid chopped or sloppy text selection. Try to capture the relevant words and avoid the irrelevant ones.
- **Step 4. Review your classifications**. After reading the narrative and reviewing your annotations, you will have a clear idea about the "big picture" of the story. Use that information to review your classification in the gray area on top of the narrative.
  - Do not select main categories (e.g. "Suspect"). If you originally selected a main category, please delete it and focus on the right subcategory.
  - Remember to conduct your classification using only the subcategories that provide more information about the incident (e.g. "Suspect/Injured").
- **Step 5. Compare your coding decisions to those of your peers**. After you are done reviewing your own annotations and classifications, please take a look at the coding that your peers conducted on the same narrative. This will help you identify annotations and classification categories that you may have missed or coded incorrectly. Please fix your annotations and classifications accordingly.
  - If you notice any issues in the annotations and classifications that your peers conducted, please leave them a comment. Indicate the name of the coder you are referring to and be specific about the issue that you noticed.
  - If you receive a comment, please address it. Once you are done with it, please mark it as "Resolved". That will help you focus on the pending comments.

Click on the link below to see a brief video of the review process.

https://tinyurl.com/leokavideo2

8 GRADING

Providing complete and high-quality annotations is crucial to guarantee high-quality performance for ML algorithms. If we use low-quality data to train the ML algorithm, the output will be filled with errors. The logic is very simple: "garbage in, garbage out." Since we are trying to help the FBI conduct their operations with a high-quality ML algorithm, there is low tolerance for poor-quality annotations in this coding exercise.

Task 1 - LEOKA Coding. The objective of the first task is to ensure that all coders complete their annotations. Coders will be evaluated based on the proportion of completion of the assigned annotations. Students who finish coding all 130 LEOKA narratives will receive 100% of the grade for this task, which corresponds to 4 points. Coders who do not finish coding their annotations will be evaluated based on the proportion of narratives completed. For example, students who completed 65 narratives, that is 50% of the total allocation, will receive 50% of the assignment grade, which is equivalent to 2 points.

Task 2 - LEOKA Validation. The objective of the second task is to ensure that we have high-quality annotations. Completing the annotations does not necessarily mean that the job was properly done. We are humans and we are all prone to error. However, the quality of information is crucial to generate high-accuracy ML algorithms. If we provide low-quality examples filled with errors and inconsistency, then the computer will "learn" from confusing information and the ML results will be inaccurate. For this reason, we need to make sure the annotations are as consistent as possible to minimize errors.

Coders will have the opportunity to revise their annotations by comparing their own coding to the decisions made by other coders in their team. The objective is to reach a high level of agreement between coders. This will give us confidence that students are applying the coding decisions in a uniform manner.

The convention to consider human annotations as high-quality is that they report an inter-coder reliability of 75% or higher. The standard metric to assess the degree of agreement is the F1 score, which is a combination of two metrics: recall and precision. The F1 score

ranges from 0 to 1, or 0% to 100%, where 0 indicates perfect disagreement and 1 indicates perfect agreement between coders.

The grade for the review assignment will be based on the curved inter-coder reliability score between coders. Since human coders rarely get 100% agreement, the inter-coder reliability score will be curved up to 100%, as long as the top inter-coder reliability score is above 75%. Label Studio automatically calculates the degree of agreement between coders as shown in Figure 16. For example, if a pair of coders in a LEOKA team has the highest inter-coder reliability score of 81%, then that top score will be considered as 100% of this assignment for a grade of 4. If another coder in that same LEOKA team did a relatively less precise job processing the narratives and the inter-coder reliability shows a score of 67%, then the curved grade for that student will be $4 \times 67/81 = 3.01$.

Coding performance at 50% or lower will not be curved up in the grading for this assignment. Coding performance at or below 50% indicates that the quality of coding is worse than flipping a coin to randomly get the correct coding by 50%-50% chance.

Task 3 - LEOKA Gold Standard Records - as extra credit. The top performing students in each LEOKA team will be invited to earn 4 extra credits for assigning annotations as "Gold Standard Record" or "Ground Truth" to be used in the ML application. This opportunity is merit-based as we need to ensure that only the best coders contribute to identifying the high-quality annotations that we are going to use for the ML algorithm. This will help us to ensure that no low-quality assessments contaminate the high-quality information required for the ML analysis.