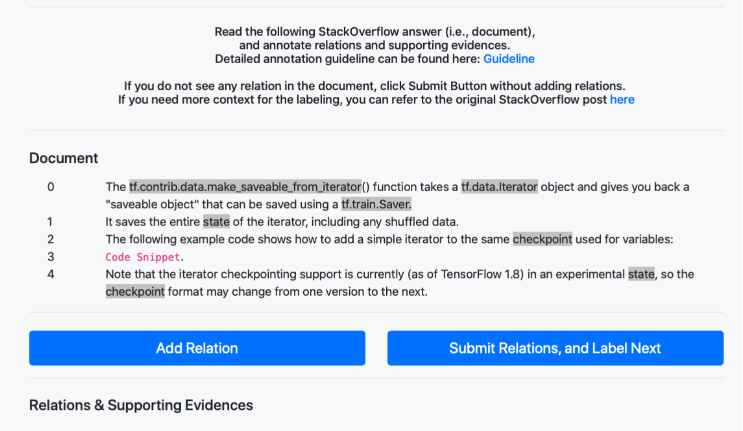
1. Workflow Overview

1. Read the annotation guideline, and set up the annotation tool [details]
2. For each document,
    a. Read the full document, while focusing on ML keywords that are highlighted in the document.
    b. Find and annotate relations of ML keywords: look for *is_compared_with* relations involving at least one of the keywords [details], and annotate them [details]
    c. Find and annotate supporting evidence: for each relation annotation, find all the sentences describing the relational fact (i.e., supporting evidence) [details].

2. Annotation Tool Set-up

1. First, setup localhost on your machine if you haven't already.
    a. If you have not done this step before,
        1. [mac] refer to this page, and follow the instructions until the chapter "Create Sites Folder": https://websitebeaver.com/set-up-localhost-on-macos-high-sierra-apache- mysql-and-php-7-with-sslhttps
        2. [Windows 10] refer to this page, and follow the chapter "Configuring PHP on IIS": https://stackify.com/how-to-host-php-on-windows-with-iis/

2. Unzip the tool at localhost root
    i. if you followed the link above, /Users/[username]/Sites/ for mac, and C:\intepub\wwwroot\ for Windows 10.
3. Open your web browser, and go to http://localhost/REAnnotationTool/re_index.php?workerId=annotator1&taskId=0&documentId=0
4. You'll see a document and buttons like below.



Read the following StackOverflow answer (i.e., document),
and annotate relations and supporting evidences.
Detailed annotation guideline can be found here: Guideline

If you do not see any relation in the document, click Submit Button without adding relations.
If you need more context for the labeling, you can refer to the original StackOverflow post here

Document

0 The tf.contrib.data.make_saveable_from_iterator() function takes a tf.data.Iterator object and gives you back a "saveable object" that can be saved using a tf.train.Saver.

1 It saves the entire state of the iterator, including any shuffled data.

2 The following example code shows how to add a simple iterator to the same checkpoint used for variables:

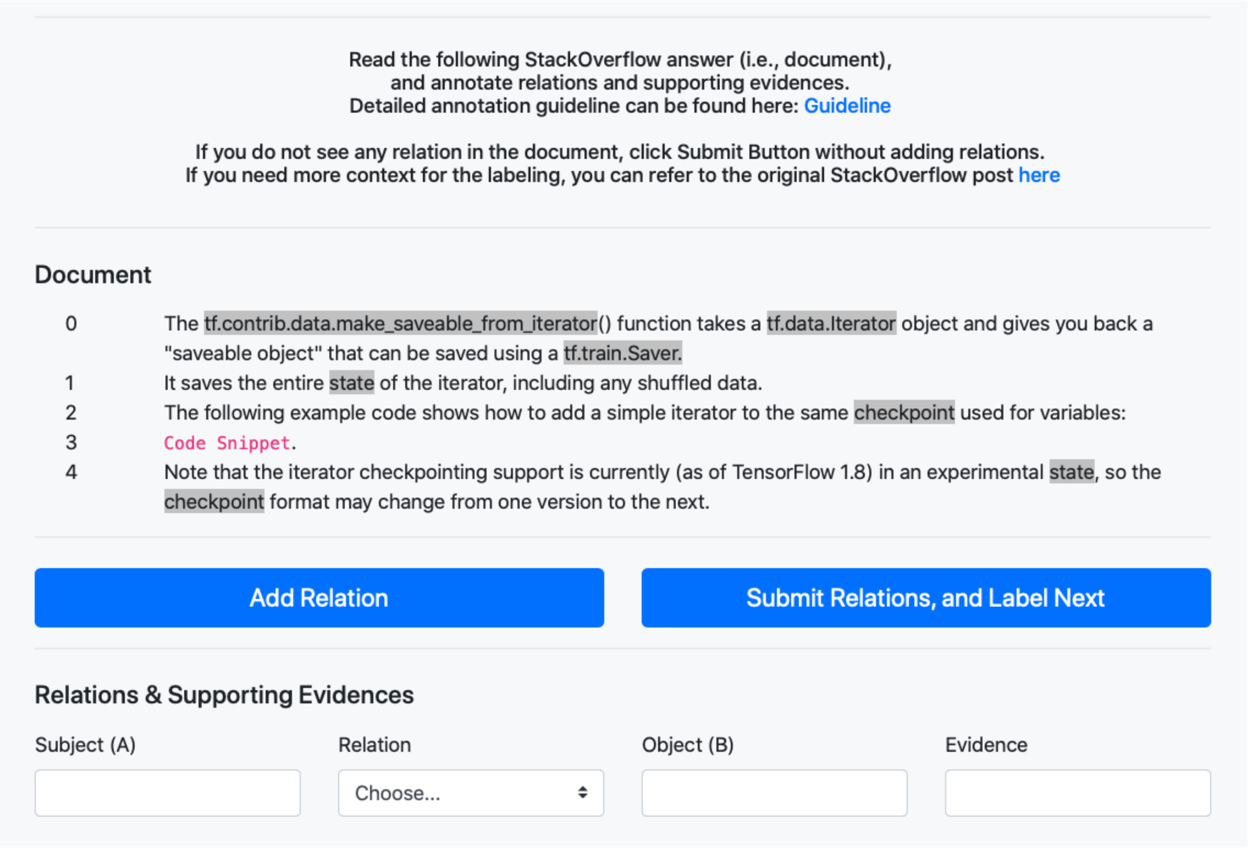
3 `Code Snippet`.

4 Note that the iterator checkpointing support is currently (as of TensorFlow 1.8) in an experimental state, so the checkpoint format may change from one version to the next.

Add Relation
Submit Relations, and Label Next
Relations & Supporting Evidences

5. Try annotating one document, and click the Submit button. The tool will save your annotation under the “re_labeling_result” folder.
   i. If the tool has successfully saved your annotation, now you are good to start annotation.
   ii. If you don’t see any new text file generated under the folder, contact the researcher for troubleshooting.

3. Annotation



Read the following StackOverflow answer (i.e., document),
and annotate relations and supporting evidences.
Detailed annotation guideline can be found here: Guideline

If you do not see any relation in the document, click Submit Button without adding relations.
If you need more context for the labeling, you can refer to the original StackOverflow post here

Document

0 The tf.contrib.data.make_saveable_from_iterator() function takes a tf.data.Iterator object and gives you back a "saveable object" that can be saved using a tf.train.Saver.

1 It saves the entire state of the iterator, including any shuffled data.

2 The following example code shows how to add a simple iterator to the same checkpoint used for variables:

3 `Code Snippet`.

4 Note that the iterator checkpointing support is currently (as of TensorFlow 1.8) in an experimental state, so the checkpoint format may change from one version to the next.

Add Relation
Submit Relations, and Label Next
Relations & Supporting Evidences
Subject (A)
Relation
Choose...
Object (B)
Evidence

a. Before you annotate anything, please read the whole document focusing on the ML keywords highlighted in the document.

i. “Code Snippet” in the document indicates places where the original Stack Overflow post had a (here omitted) code snippet.

ii. If you need more context, you can refer to the original Stack Overflow post. You can open the post by clicking the link (“here”) at the top of the page.

b. When you’re ready to annotate, click the “Add Relation” button or press “esc”. Then you’ll see text input boxes and a dropdown menu for the relation annotation.

i. Copy and paste keywords into *Subject (A)* and *Object (B)* text boxes.

1. For each relational fact you annotate, **at least one input (subject/object) should be a highlighted ML keyword** in the given document.
2. Copy and paste the keywords as is. Make sure to input the terms **as shown in the given document**. Please do not paraphrase or remove words in the middle.
3. If you no longer want to add a relation, leave the text box empty, and select *no relation* from the relation dropdown menu.
4. Spaces at the beginning/end of the terms will be removed in the post-processing steps, so feel free to submit them as is.

ii. Choose a relation that holds between *Subject (A)* and *Object (B)* from the “Relation” dropdown menu.

1. If you change your mind while inputting relations (e.g., you want to remove the relation you were annotating), select *no relation* from the relation dropdown.

iii. Input the sentence identifiers (numbers on the left side of sentences) into the *Evidence* text box. If you find multiple supporting evidences, separate them using “,” like “0,1,2”.

iv. If you have more relations to annotate, click the *Add Relation* button again, or press “esc” to add new input fields.

v. Read the document again to make sure that you don’t miss any relation in the document.

a. Once you annotate all the relations you found, click “Submit Relations, and Label Next” button to move to the next document.

4. Annotate Relations

Identify relations, *is compared with*, of ML keywords that are highlighted in the given document. If you cannot find any relational facts of ML keywords, click the *Submit Relation* button.

- **Relation Examples**
    - A ***is compared with*** B
        - Sigmoid (A) is compared with softmax (B)
        - LSTMBlockCell (A) is compared with BasicLSTMCell (B)

- **Annotation Notes**
    - **Note1 Relations must be reflected in the document**, without relying on external world knowledge (e.g., what you learned from your coursework). Do not label relations that you can only infer by the structure of a document (e.g., listed together).
    - **Note2** Make sure to input the terms as shown in the given document. Please do not paraphrase or remove words in the middle. If a term is in a plural form, input that in plural form, and do not change it to a singular form. Do not introduce new words, but just input the nouns given in documents.
    - **Note3** Relations could be spread over multiple sentences.

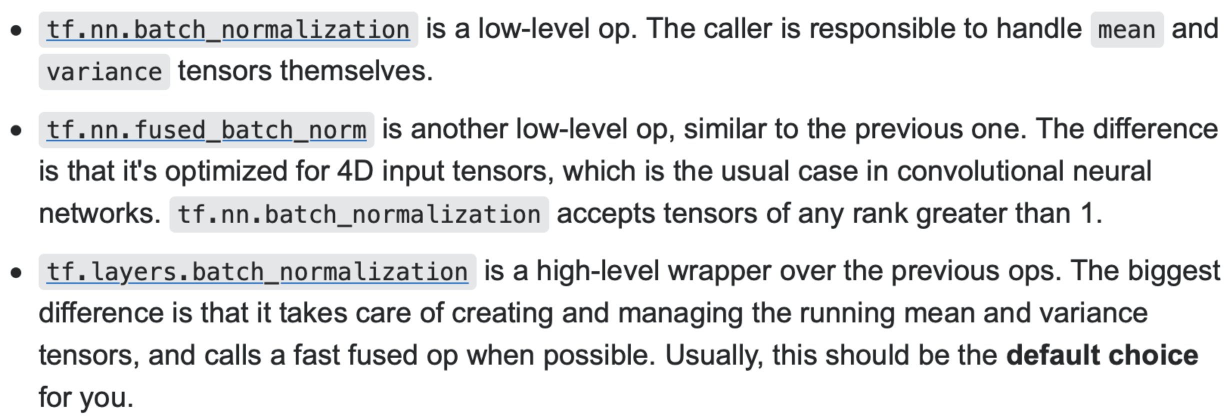
- **Relations**
    - A ***is compared with*** B: is a symmetric relation. You can label two terms with this relation when the document is comparing the two (see examples below), such as A *is an alternative to* B, A *is better/worse than* B, A *should alter* B, and A *is similar to* B. You can usually find this relation when the document is comparing two, or suggesting to alter one with another. The document should describe how or why the two are different, or similar.

    Annotation Example 1

    - **[Given document]** The main difference between tf.contrib.rnn.LSTMBlockCell and BasicLSTMCell is that BasicLSTMCell is implemented in terms of primitive TensorFlow ops, so it's useful code to copy and edit if you want to play with your LSTM architecture.
    - **[Annotation]** tf.contrib.rnn.LSTMBlockCell *is compared with* BasicLSTMCell
    - **[Explanation]** The given document describes the difference between the two methods. Although BasicLSTMCell is not highlighted in the document, as long as one of the terms is highlighted, you can still annotate the relation.

    Annotation Example 2

    - **[Given document]**



- `tf.nn.batch_normalization` is a low-level op. The caller is responsible to handle `mean` and `variance` tensors themselves.
- `tf.nn.fused_batch_norm` is another low-level op, similar to the previous one. The difference is that it's optimized for 4D input tensors, which is the usual case in convolutional neural networks. `tf.nn.batch_normalization` accepts tensors of any rank greater than 1.
- `tf.layers.batch_normalization` is a high-level wrapper over the previous ops. The biggest difference is that it takes care of creating and managing the running mean and variance tensors, and calls a fast fused op when possible. Usually, this should be the **default choice** for you.

- **[Annotation]**

tf.nn.batch_normalization *is compared with* tf.nn.fused_batch_norm
tf.nn.batch_normalization *is compared with* tf.layers.batch_normalization
tf.nn.fused_batch_norm *is compared with* tf.layers.batch_normalization

- **[Explanation]** You can infer that they are being compared based on the structure, which you need to be careful annotating this, as per Note1. However, it is safe to annotate them in this document because they are also being compared explicitly in text, by “similar to the previous one” or “The biggest difference is that”. When there are more than two that are compared, you need to annotate all of the combinations.

5. Supporting Evidence

For each relational fact you annotated, select **all** sentences that support the relation (e.g., sentences explaining the differences between the two terms) instances as supporting evidence.

[Example document]
0: The issue stems from tf.zeros.
1: Initializing your weights to zero is a common mistake when defining a neural network.
2: This article explains the reasoning behind it
3: very approximately, all the neurons will have the same value, so the network won't learn.
4: Instead you should initialize your weights to small random numbers, and a typical scheme is to use tf.truncated_normal() with a standard deviation inversely proporational to the number of input units.

[Relation annotation you find] tf.zeros *is compared with* tf.truncated_normal
[Supporting evidence] 0, 1, 3, 4

[Wrong supporting evidence] 0, 4
[Explanation for wrong annotation] You need to annotate ALL sentences that discuss the reason why tf.truncated_normal should alter tf.zeros.