TLDR: it seems too difficult for the model to distinguish between MISC and other labels

Original performance results:

```
Overall F1 score (stringent): 0.6813186813186813
Overall F1 score (ignoring entity type): 0.7424325811777654
0.746

All metrics:
{'LOC': {'f1': 0.7552083333333333,
         'number': 550,
         'precision': 0.7225913621262459,
         'recall': 0.7909090909090909},
 'MISC': {'f1': 0.0, 'number': 310, 'precision': 0.0, 'recall': 0.0},
 'ORG': {'f1': 0.5919439579684763,
         'number': 263,
         'precision': 0.5487012987012987,
         'recall': 0.6425855513307985},
 'PER': {'f1': 0.791537025513379,
         'number': 785,
         'precision': 0.7737226277372263,
         'recall': 0.8101910828025478},
 'overall_accuracy': 0.9520377456625924,
 'overall_f1': 0.6813186813186813,
 'overall_precision': 0.7159353348729792,
 'overall_recall': 0.649895178197065}
```

Clearly the model isn't making any predictions for 'MISC' entities. I tracked the problem down to the `id2label` dictionary we use to convert indexes to labels (quick demonstration below):

The model returns a list called `pred` that looks something like this: [11.443, 8.592, 3.401, -4.300, -5.294, -5.920, 6.482, 8.149, 0.1893]. Then the index of the max value (11.443) is 0. Using the default `id2label` dictionary, this corresponds to an 'O' label.

Our evaluation notebook was using the default dictionary:

However, when I checked the config.json file for Davlan, the dictionary looked like this:

Essentially, I thought that the model might have been coming up with correct numerical predictions for all 4 labels (‘PER’, ‘ORG’, ‘LOC’, ‘MISC’) and simply misassigning the labels due to an incorrect dictionary (the default dictionary used the ‘DATE’ label instead of the ‘MISC’ label). I ran evaluation again using the new `id2label` dictionary and got these results:

```
Overall F1 score (stringent): 0.030530583465538517
Overall F1 score (ignoring entity type): 0.07142070686681104

All metrics:
{'LOC': {'f1': 0.001113275814082939,
         'number': 550,
         'precision': 0.000569082557619609,
         'recall': 0.025454545454545455},
'MISC': {'f1': 0.029654036243822075,
          'number': 310,
          'precision': 0.030303030303030304,
          'recall': 0.02903225806451613},
 'ORG': {'f1': 0.019047619047619046,
         'number': 263,
         'precision': 0.025477707006369428,
         'recall': 0.015209125475285171},
 'PER': {'f1': 0.3712848651120255,
         'number': 785,
         'precision': 0.289586305278174,
         'recall': 0.5171974522292994},
 'overall_accuracy': 0.0343277482631794,
 'overall_f1': 0.030530583465538517,
 'overall_precision': 0.016366179083040406,
 'overall_recall': 0.22693920335429768}
```

This run was actually much worse. I ran through the comparisons file to check why. It seems that the default `id2label` is much more accurate than the one in the config file; 5 should indeed by ‘B-ORG’ and 6 is almost always ‘I-ORG’, etc. I’m not sure why the config file has an `id2label` dictionary that it doesn’t seem to use, but didn’t have time to investigate.

Rather than change the other labels, I just changed the values 1 and 2 in the dictionary to be ‘MISC’ instead of ‘DATE’, but got the same results and 0 predictions for MISC.

When I went into the comparisons file, I ran into three main categories of errors regarding ‘MISC’ categorizations (NOTE: ground truth labels on left, predictions on right):

1. **Human labeling error**

- In this case, it's clear that the United Japan Society is actually a noun that is the direct object of Eisenhower's cancellation. It should be an organization, and the model correctly predicted that (5, 6 correspond with 'B-ORG', 'I-ORG' model predictions on the right). The human labeler incorrectly labeled them 'MISC', perhaps confused by the 'United' descriptor.
- These errors are relatively less common

2. **Niche 'MISC' entities and nationalities**

d

- In the first row, it seems the model doesn't recognize historical 'MISC' entities (specifically plans, programs, documents, and doctrines), probably because they contain words that are used everyday (like 'Iron' or 'four point plan'). It has an easier time with names, common locations, and organizations, which contain easily recognized words like 'society' or 'city'.
- In the second row, the model doesn't recognize nationalities like 'Japanese' or 'European', although it will recognize locations like 'Japan' and 'Europe'.
- The model is completely missing these, not just mis-labeling them, meaning that it views them as closer to 'O' labels (index 0, according to the dictionary) than any 'ORG', 'PER', or 'LOC' label.
- This also indicates 'MISC' labels probably shouldn't be associated with the 1 and 2 indices in the dictionary.
- NOTE: in the future, we could take a look at the lists being put out for MISC entities and check if their max values are only narrowly beating out other values. For example, this list [11.443, 0.003, 0.003, 0.003, 9.893, 0.003, 0.003, 0.003] would be more promising than [11.443, 0.003, 0.003, 0.003, 0.003, 0.003, 0.003, 0.003], because it would indicate that the model also thinks the word could be an 'ORG' entity and may be able to improve with larger training samples.

3. **Confusion with other labels**

- The exceptions to the second error are instances where the ‘MISC’ entity is spelled exactly or almost exactly like an entity that otherwise would be labeled ‘PER’, ‘ORG’, or ‘LOC’.
- ‘US’ is the biggest instance of this error. The model doesn’t seem to be able to distinguish the nuance between the two uses (as a noun/’LOC’ and a descriptor/’MISC’). It may require us to revise our definitions of the labels and possibly get rid of the ‘MISC’ label. Maybe a model that does better at distinguishing POS (parts of speech) would be better at this if given larger training samples.

As a last check, I wanted to see if the max values for these misclassifications were split. For example, perhaps the model is lumping all instances of ‘US’ into ‘LOC’ and returning an index of the 7 or 8 (per `id2label`). However, the max values themselves could be different; maybe ‘LOC’ values cluster around 3 and ‘MISC’ values cluster around 8. Then we could differentiate ‘MISC’ entities by looking at values rather than indexes and we would want the histogram of max values to look like this:

I filtered max values by label and plotted histograms: