In this lab we will explore the implementation of a Sructured Perceptron for Named Entity Recognition. The perceptron needs to learn to predict one of 5 labels ['O', 'LOC', 'ORG', 'MISC', 'PER'] instead of two like we saw in the first lab. For that we need to make use of the structure in the data, which will be given by our feature representation. The implementation takes around 6 minutes to complete.

Word-Tag Feature Representation

For the Word-Tag feature, a function **phi_1(x, y, cw_cl_counts)** was implemented which returned the counts of word-tag pairs (x and y) that were present as keys in the total Word-Tag counts seen in training (cw_cl_counts), this is to ensure that we only update our weights on counts of pairs that we have seen in our train dataset.

Our algorithm uses a function called **get_combis(x)** to generate the space of all possible combinations of labels (y) for our particular x, and get their respective feature representation using phi_1. Then we use the function **get_combis_values** to obtain the total sum of each combination with the weights.

After we have the sum of each possible combination, the function **get_pred_label** finds the highest value using argmax and returns the correspondent labels. In case of a tie between sums, the argmax will make a random choice between all the possible maximums. This is to ensure the model has no bias to select a specific label.

Finally if the predicted labels are not correct, then our perceptron updates the weights adding the values for the feature representation of the correct label, and subtracting the feature representation values for the incorrect labels.

After 5 passes with randomized order we tested our weights and the result was a **micro F1 score of 0.62**.

To get the most weighted features, the results for each label were sorted by value and then by key, so they could be reproduced. This is because there could be more than 10 keys with the top "value" so each time the dictionary would give different results.

The most weighted features for each label were the following

ORG	**MISC**	**PER**	**LOC**	**O**
&	C$	A.	ABABA	14,775,000
ABC	162	A.de	ABDERDEEN	1.09
AD-DIYAR	AMERICAN	Aamir	ABIDJAN	10,650,407
AD	American	Adrian	ADDIS	11.38
AHOLD	Argentine	Affleck	AIRES	17-16
AHRONOTH	Australian	Ahmed	AJACCIO	1995
AL-ANWAR	Austrian	Akam	AKRON	290
AL-WATAN	Baseball	Akram	AL-MUNTAR	3024.95
AL	Bedi	Alan	ALKHAN-YURT	4
AN-NAHAR	Belgian	Alastair	ALLENTOWN	50-75

From what we can see these features make sense for each class, in ORG we have AD–DIYAR, AL-ANWAR, AL-WATAN which are all newspapers in Arabic countries (organizations). In MISC we have mostly demonyms like American, Austrian, Australian. In PER we have names and last names like Affleck, Adrian, Ahmed, Alan. In LOC we have places like ABDERDEEN, ADDIS ABABA, or parts of places like AIRES (for Buenos Aires). Finally for O we have mostly numbers.

Word-Tag + Tag-Tag Feature Results

In this section, our perceptron will include a new feature (Tag-Tag) which simply is a representation of the current Y label and the previous Y label. This gives more context to the perceptron to identify possible labels, for example it might be more likely for a PER label to come after a previous PER label (name & last name).

After 5 passes with randomized order we tested our weights and the result was a **micro F1 score of 0.71**.

The most weighted features for each label were the following:

ORG	**MISC**	**PER**	**LOC**	**O**
Newsroom	Polish	Fogarty	England	2
Oakland	Dutch	Slight	Finland	3
CHICAGO	English	Tortelli	LONDON	AT
Milwaukee	French	Armstrong	PARIS	Division
Orient	GMT	Camerlengo	Pakistan	Tuesday
Seattle	GTR	Capiot	WASHINGTON	out
AN-NAHAR	German	Corser	AMSTERDAM	)
AS-SAFIR	Scottish	Kocinski	Australia	-
Atletico	C$	Koerts	BRUSSELS	1-1
BALTIMORE	CENTRAL	M.	CITY	1

We can see that for the MISC, PER and LOC labels the identification is very accurate and similar to the previous results (demonyms, names and cities). In the ORG labels we see many labels that could be LOC like Oakland, Chicago, Seattle, but after exploring the training dataset we see all of these were labeled as ORG and were probably scores for some sport (CHICAGO 2 Milwaukee 0 — ORG O ORG O, for example). It is likely that these labels are more represented as ORGs than LOC and that is why they appear in this class.

The difference in F1 score between both models is around 0.09, which is an increase of 14% compared to using only one feature. This difference is expected because with two features our perceptron can use more contextual information and learn the structure of the data in a more accurate way. The tag-tag feature helps our perceptron disambiguate some instances where it was not sure which label was correct.