

ANLP Assignment 2 2024

Marked anonymously: do not add your name(s) or ID numbers.

Use this template file for your solutions. Please start each question on a new page (as we have done here). Do not remove the pagebreaks. Different questions may be marked by different markers so your answers must be self-contained.

1 Logistic regression tagging with word embeddings

Comparing the two models, we see that all three types of averages (micro-average, macro-average, and weighted average) for all metrics reveal better scores for the Logistic Regression Model, almost doubling the scores from the most frequent tag model. A significant improvement across these three averages indicates that the Logistic Regression Model performs better at assigning the correct BIO tag to a token. Take as an example the weighted average for Precision, Recall, and F1-score; the most frequent tag model yields the following values: 0.32, 0.22, 0.24, respectively, while the Logistic Regression Model gives 0.53, 0.49, 0.50.

Considering the **B-person_name** tag, we provide examples in Table 1. We observe that the frequent tag model performs poorly on words with the **B-person_name** tag, assigning the **O** tag to all of them, as depicted in Table 1a. This reveals one of the models' weaknesses, which we discuss further below. On the other hand, the Logistic Regression Model successfully identifies the correct tag for some tokens in Table 1b, leading to more accurate recognition of names, even though it occasionally introduces some false positives.

Words	Truth Set	Predicted Set	Words	Truth Set	Predicted Set
bruggeman	B-person_name	O	bruggeman	B-person_name	B-date_to
shaida	B-person_name	O	shaida	B-person_name	B-person_name
delana	B-person_name	O	dalana	B-person_name	B-person_name
jon	O	O	jon	O	B-person_name
jacuzzi	O	O	jacuzzi	O	B-person_name
mural	B-person_name	O	murad	B-person_name	O

(a) Most frequent Tag model.

(b) Logistic Regression model.

Table 1: Labels and predicted tags for each word for both models.

Analyzing the most frequent tag model, we observe that it struggles to assign correct tags to unseen words (words that appear in the validation set but were absent in the training set). This issue arises because the probability of tags given these unseen words is zero, as the word-tag pair does not exist in the training set. As a result, the model defaults to assigning these words the **O** tag. This behavior is explained by the maximum likelihood estimate (MLE) formula:

$$P(t | w) = \frac{\text{count}(t, w)}{\text{count}(w)}$$

Where $\text{count}(t, w)$ is the number of occurrences in the training data set such that a token w appears with tag t . This value is zero for unseen words, regardless of the tag; meaning that $P(t|w)$, the probability of assigning tag t given a word w , will also be zero. As a result, the model assigns the **O** tag to these words.

On the other hand, the logistic regression model makes its predictions based on word embeddings, as opposed to the words themselves. Since $P(t | w)$ is now a logistic regression model, we have that $P(t | w) \in (0, 1)$ which is never zero, avoiding the previous problem.

Additionally, embeddings provide more information by capturing semantic meaning of words depending on the context. This allows the model to link unseen words to similar ones from the training set, improving its ability to correctly assign tags based on probabilities calculated for each embedding. This approach reduces the number of words being misclassified with the **O** tag, as seen in Table 1a. Overall, we can see that the Logistic Regression Model performs better than the most frequent tag Model.

2 BIO tagging for slot labeling

The `predict_bio_tags()` function generates all possible tags for each token based on word embeddings, sorted by their probability. It then retrieves the most probable tag and checks if it meets the BIO tagging constraint. Specifically, if the tag starts with the prefix **I-**, the function compares its type with the type of the previous tag. If they don't match, the function moves to the next most probable tag. Once a tag satisfies the constraint, it is then assigned to the token. In other words, since BIO tags begin with **I-** or **B-**, we are ensuring that every **I-** follows a **B-** or **I-** tag of the same type (BIO constraint). This procedure is repeated for each token in the input sentence.

Word	<code>predict_independent_tags()</code>	<code>predict_bio_tags()</code>
5	B-date_period	B-date_period
nights	I-date_period	I-date_period
from	O	O
15th	B-date_from	B-date_from
of	I-date_to	I-date_from
October	B-date_from	B-date_from
?	O	O

Table 2: BIO Predictions using `predict_independent_tags()` and `predict_bio_tags()`.

Observing Table 2, the model using `predict_independent_tags()` assigns the tag **I-date_to** to the token **of**. However, **I-date_to** does not align with the BIO constraint (the tag type **date_to** does not match with the tag type **date_from** of token **15th**), so `predict_bio_tags()` replaces **I-date_to** with **I-date_from**, which is the most probable tag that satisfies the constraint.

As a suggested approach, the Viterbi algorithm is well-suited because it uses dynamic programming method to find the most likely sequence of tags in a Hidden Markov Model (HMM) given an observation sequence. In a HMM, the probability of each tag depends only on the previous tag (Markov Assumption), allowing us to enforce BIO constraints by defining valid and invalid transitions. Specifically, if the current tag (state) begins with **I-**, the previous tag (state) must begin with **B-** or **I-** of the same type, for these transitions, the probability is equally distributed to all valid transition from the previous tag; otherwise, the transition probability is set to zero. Additionally, emission probabilities are given by $b_s(o_i) = \frac{P(t|o_i)P(o_i)}{P(t)} \propto \frac{P(t|o_i)}{P(t)}$, where o_i is the i^{th} token of the sentence, and t denotes the tag of state s ; $P(t|o_i)$ is given by the Logistic Regression Model, and we may use Maximum Likelihood to estimate $P(t)$. At each step, the algorithm uses both transition and emission probabilities to compute the most likely path with a correct BIO tagging format by skipping invalid transitions.

The computational complexity is $\mathcal{O}(S^2 \times T)$, where S is the number of tags and T is the sentence length. The complexity comes from calculating each possible tag transition (from each previous tag to each current tag) at each position in the sentence. In contrast, the complexity of the provided algorithm is $\mathcal{O}(S \times T)$, in the worst scenario, as it checks all the possible tags for each token. However, the typical complexity is $\mathcal{O}(T)$, since the most probable tag often complies with the BIO taggings rule.

Overall, the provided algorithm is more efficient and requires less memory, making it suitable for longer sequences. Unlike Viterbi algorithm, it doesn't store backpointers, which Viterbi uses to trace back from the last token to determine the optimal tag sequence.

Finally, the Viterbi algorithm provides more accurate results by considering dependencies between tags across the sequence. In Table 2, Viterbi would likely help the model recognize **October** as part of the span **15th of October** and correctly assign it the tag **I-date_from**.

3 Error Analysis and Feature Engineering

We identified that the current model frequently misclassifies the span label **date_from**. We examined this issue closely, presenting specific examples along with reasoning as to why these errors might occur and suggesting potential ways to address them.

Considering this, the model sometimes fails to recognize tokens, which should be assigned to date-related tag labels, such as **date**, **date_from**, and **date_to**. In the following example "A double room for tomorrow for 8 days", the model does not tag **tomorrow** as date-related; instead, it incorrectly marks it as **O** (outside tag), as shown in Table 3 below. This issue arises because the embeddings may not provide enough information to identify the type of entity of each word, as indicated by the **Entity Type** column in Table 3.

Word	Entity Type	BIO Annotations	Original BIO Predictions	Updated BIO Predictions
a	-	B-rooms	B-rooms	B-rooms
double	-	I-rooms	I-rooms	I-rooms
room	-	I-rooms	I-rooms	I-rooms
for	-	O	O	O
tomorrow	DATE	B-date_from	O	B-date
for	-	O	O	O
8	DATE	B-date_period	B-date_period	B-date_period
days	DATE	I-date_period	I-date_period	I-date_period

Table 3: Comparison between BIO Predictions of the Original and Updated Logistic Regression Model incorporating the linguistic feature entity type of each word.

To reduce the problem, we decided to add the linguistic feature of **ent_type** to the embeddings. We believe this approach will be effective because **token.ent_type** frequently recognizes common entity types such as dates. This provides the model with clearer indications that certain words, such as **tomorrow** should be assigned with a date-related tag. The results of the updated model in Table 3 show that **tomorrow** is now assigned to a date-related tag, **B-date**.

However, the model still struggles to accurately distinguish between similar span tags, such as **date**, **date_from**, and **date_to**. This challenge occurs because each tag has a specific contextual meaning that often depends on surrounding words. For instance, the tag **date_from** usually relies on words such as **from** or **starting**, while **date_to** might depend on words **to** or **until**.

Word	Token Head	BIO Annotation	Original BIO Predictions	Updated BIO Predictions
Two	nights	B-date_period	B-date_period	B-date_period
nights	nights	I-date_period	I-date_period	I-date_period
from	nights	O	O	O
Friday	29th	B-date_from	B-date_from	B-date_from
29th	from	I-date_from	B-date_to	I-date_from

Table 4: Token’s Head, BIO Annotations, and BIO Predictions of the Original and Updated version of the Logistic Regression Model.

Analyzing the sentence "Two nights from Friday 29th", as shown in Table 4, the token **29th** should be labeled as **date_from**, based on the context provided by the word **from**. But the original model incorrectly assigns the tag type **date_to** to **29th**, interpreting it as the end of the date range rather than part of a starting date. This misclassification occurs because the current

embeddings do not capture the relationship between **29th** and its neighboring word **from**. The existing embeddings represent each token individually and may lack information about this kind of contextual links between tokens.

To improve the model’s performance and address this tagging issue, we further added an additional linguistic feature, **token.head** (syntactic parent of the token), to enrich even more the embeddings’ information. By concatenating **token.head.vector** to the previous modified embeddings, this feature provides more context about how each word in a sentence connects to others, especially highlighting which words act as **heads** that influence the meaning of neighbouring words.

In Table 4, **token.head** offers meaningful information because dates are often surrounded by words such as **from**, **to**, or **until** that indicate a time range. In the following sentence "**Two nights from Friday 29th**", the token head feature leads the model to recognize that **29th** is linked to the token **from**, as illustrated in Figure 1. With this addition, the model can correctly assign the token **29th** to the tag type **date_from** rather than **date_to**, as reflected in the updated predictions in Table 4.

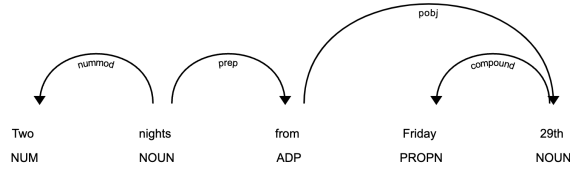


Figure 1: Visualization of dependency parse, showing the head relationships for each token.

In summary, the updated Logistic Regression Model, enhanced with embeddings that incorporate additional linguistic features, demonstrates an improvement over the original. The average metrics across all slot labels show higher values, as shown in Table 5, indicating the benefits of including features, such as head dependencies and entity types. Additionally, we noticed that the three metrics (Precision, Recall, and F1-score) of the slot label **date_from** are significantly better increasing from 0.44, 0.55, 0.49 in the original model to 0.62, 0.65, 0.63 of the updated model, respectively.

	Original Model			Updated Model		
Metric	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Micro Avg	0.47	0.47	0.47	0.56	0.54	0.55
Macro Avg	0.40	0.41	0.40	0.45	0.45	0.44
Weighted Avg	0.46	0.47	0.46	0.55	0.54	0.53

Table 5: Comparison between the averages of Precision, Recall, and F1 Score for Original and Updated Models.

Going further, we observe that the modifications made to our model yield promising results across most slot labels. For example, in the sentence “**From the 16th to the 30th, a single room for 2 adults and 1 child**”, the updated model correctly assigns the slot label **rooms** to the tokens **a**, **single**, and **room**, identifying the word **room** as their syntactic head. In contrast, the original model incorrectly labeled these tokens as **0**. By incorporating head dependency, the model is better equipped to recognize when tokens belong to the same span, reducing misslabeling errors. This improvement makes the model to perform better for similar cases, such as **2 adults** or **8 people**, labeling those correctly through their syntactic relationship (the head of **2** is **adults** and **people** is the head of **8**).

4 Final Test Reporting and Reflection

Starting with the comparison of the original Logistic Regression model on both the validation and test sets, we observed from the classification report that the model performs better on the validation set, as shown by the average metrics, which provide an overall assessment of model performance. For example, the micro averages for Precision, Recall, and F1-score are 0.37, 0.39, and 0.38 on the test set, respectively, compared to 0.47 for all metrics on the validation set. This pattern is similar in our model, with micro average metrics of 0.43, 0.44, and 0.43 for Precision, Recall, and F1-score, respectively, on the test set, while the validation set scores were higher at 0.56, 0.54, and 0.55.

Since we didn't conduct hyperparameter optimization on the original Logistic Regression Model using the validation set, we cannot explain with certainty the performance discrepancy between the validation and test sets. Given that the training, validation, and test sets were randomly sampled from the same source, we would expect similar performance between validation and test sets. However, if the validation set happens to be more closely aligned with the training data, it may yield higher performance scores. On the other hand, the test set could contain unique or less frequent patterns, leading to lower performance metrics.

Unlike the original Logistic Regression model, our updated model was refined based on results and examples from the validation set with additional linguistic features, such as syntactic head of a token and its entity type. This helps the model recognize patterns in the validation data and assign correct tags to tokens. However, we observed that these patterns are less frequent in the test set, leading to a lower performance on the test set than on the validation set.

For instance, our model, with information about the syntactic head of a token, is able to assign the correct tag to dates when their syntactic head is **from** or **until**. The same happens to numbers when their surrounding words include **rooms**. As shown in Table 6, the updated model assigns the correct tag type to **4**, **23rd**, and **May**, since their syntactic head is **rooms**, **from**, and **until**, respectively. However, these patterns are less frequent in the test set. For example, the pattern "**from** [DATE]" appears only twice in the test set compared to seven times in the validation set, while the support for **date_from** label is similar. This difference in frequency leads the model to perform worse on the test set.

Overall, the modified model with additional linguistic features performs better than the original logistic regression model across the classification metrics on both the validation and test sets. This is because examples similar to the one above are still present in the test set, even though they are less frequent.

Word	Token Head	BIO Annotation	Original BIO Prediction	Updated BIO Prediction
4	rooms	B-rooms	B-adults	B-rooms
single	rooms	O	O	O
rooms	rooms	O	O	O
from	rooms	O	O	O
the	23rd	O	O	O
23rd	from	B-date_from	B-date_to	B-date_from
until	rooms	O	O	O
the	May	O	O	O
29	May	B-date_to	B-kids	O
May	until	I-date_to	O	B-date_to

Table 6: Comparison between BIO predictions of the Original and Updated Logistic Regression Model for the sentence **4 single rooms from the 23rd until the 29 May**.