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

After examining the classification reports of both the logistic regression and most frequent tag models, we can observe significant differences in their performance on various BIO tag types. Specifically, as shown in Table 1, the 'B-kids' tag type shows a notable improvement in the logistic regression model over the most frequent tag model.

Table 1: Comparison of B-kids Tag Classification Reports

Tag	**Logistic Regression**			**Most Frequent**			**Support**
	Precision	Recall	F1-score	Precision	Recall	F1-score	
B-kids	0.71	0.83	0.77	0.00	0.00	0.00	6

To understand why the logistic regression model outperforms the most frequent tag model on the 'B-kids' tag, let's examine specific examples from the models' full report.

- **"1 adults and 8 children. Saturday to Tuesday"** In this sentence, Most Frequent model mislabels ("8 children") as **adults** instead of **kids**, and mislabels ("1 adults") as **date_period** instead of **adults**. In contrast, Logistic Regression model successfully identifies all the slots as per the annotations. We found that the Logistic Regression model demonstrates feature-based differentiation by effectively recognizing numerical expressions and their associations with specific slots. For instance, it distinguishes between numbers linked to "adults" and "kids" based on context, ensuring that quantities are assigned to the correct categories. Additionally, the model exhibits temporal understanding by identifying prepositions like "to" that indicate a range, enabling it to correctly tag dates as date_from and date_to.

- **"One bedroom, 2 adults and one minor child."** The Logistic Regression model correctly identifies the kids span (6, 7) ("one minor"). However, the Most Frequent tag model fails to predict any tag for this span. We supposed the Logistic Regression model's better performance on the **kids** tag can be attributed to several key factors. Firstly, through effective feature utilization, the model leverages word embeddings to capture semantic meanings, recognizing that words such as "child" and "minor" are related to the concept of kids. Additionally, it employs morphological features to identify that "child" is singular and "children" is plural, which aids in accurate slot identification by distinguishing between different forms of the word.

The logistic regression model performs better on the **kids** tag type because it effectively utilizes advanced features, understands contextual nuances, and learns from training data patterns. Firstly, by leveraging word embeddings, the model captures semantic meanings and recognizes that words like "child" and "minor" are related to the concept of kids. For example, in the sentence **"One bedroom, 2 adults and one minor child,"** the logistic regression model correctly identifies **"one minor"** as part of the **kids** slot, while the most frequent tag model fails to do so. Secondly, it demonstrates strong contextual understanding by considering surrounding words such as "one" and "minor," which provide cues that "child" refers to the **kids** slot, enabling accurate tagging. Lastly, the model benefits from learning patterns present in the training data where instances of "child" or "children" are associated with the **kids** slot. This learning allows it to generalize from seen examples and accurately assign the **kids** tag in new sentences. Overall, these capabilities contribute to the logistic regression model's superior performance in identifying the 'kids' tag compared to simpler models that do not utilize such comprehensive features.

2 BIO tagging for slot labeling

2.1 What does the predict_bio_tags() do, and why does it work?

The predict_bio_tags() function predicts a sequence of BIO tags for an input token sequence while ensuring compliance with BIO constraints, such as requiring an 'I-<tag>' to follow a 'B-<tag>' or 'I-<tag>' of the same type. If these constraints are violated, it adjusts the predictions accordingly. By considering the previous tag during each prediction, the function introduces a dependency that captures sequential patterns without altering the underlying model. Its greedy approach selects the most probable valid tag for each token independently, improving the efficiency while maintaining adherence to BIO rules.

2.2 Adapt Viterbi algorithm

Assumptions: To adapt the Viterbi algorithm for tag sequence selection, we define transition probabilities based on BIO constraints, assigning uniform non-zero probabilities to legal transitions and zero to illegal ones. The emission probabilities are directly taken from the model's existing $P(\text{tag} \mid \text{token})$, ensuring compliance with the BIO rules without altering the model or its estimation procedure.

To modify the Viterbi Algorithm, the process involves adapting the decoding step without altering the underlying model. During initialization, the probabilities for the first token are computed using the emission probabilities. In the recursion step, for each subsequent token, the maximum probability for each state (tag) is determined by considering the maximum probability from the previous states, weighted by the transition probabilities that enforce BIO constraints, and the emission probability of the current tag for the given token. After constructing the Viterbi matrix, the most probable sequence of tags is obtained through backtracking. Importantly, this modification avoids changing the model or its estimation procedure for $P(\text{tag} \mid \text{token})$. Transition probabilities are defined solely based on BIO constraints rather than learned parameters, ensuring compliance with the constraint of not altering the model or its training process.

2.3 Compare and contrast predict_bio_tags and Viterbi

The predict_bio_tags approach aims to select the most probable valid tag for each token based on immediate observations and constraints. This method is quick and efficient, making it suitable for applications where speed is crucial, and approximate solutions are acceptable. However, it relies on a greedy strategy, enforcing BIO constraints by skipping invalid tags and relying only on the immediate previous tag. This simplicity comes with a drawback: errors in early predictions can propagate, potentially reducing the overall accuracy. Its computational efficiency, with a time complexity of $O(N \cdot T)$, makes it ideal for large datasets or real-time scenarios, as it breaks loops early and minimizes redundant computation.

In contrast, the Viterbi algorithm seeks to find the globally optimal tag sequence that maximizes the joint probability across the entire sequence. By encoding BIO constraints into the transition probabilities, it effectively prunes invalid paths, ensuring that only legal tag sequences are considered throughout the computation. This global perspective leads to higher accuracy, as it considers the context of the entire sequence and reduces the impact of local errors. However, this comes at the cost of increased computational complexity ($O(N \cdot T^2)$), as all transitions between tag pairs must be evaluated. Additionally, its implementation involves dynamic programming tables, which can be harder to debug and maintain compared to the straightforward predict_bio_tags function.

3 Error Analysis and Feature Engineering

3.1 Analysis of the Current System: Frequently Mispredicted Labels

According to the full classification report of the current system (Logistic Regression Model), we observed that while the overall system achieves a weighted F1-score of 0.52 for BIO tags and 0.46 for slot labels, performance for some span labels remains poor. Precision and recall are particularly low for spans like rooms, time, and time_from, indicating a systemic inability to generalize effectively across all labels. For example, time_from has no successful predictions, and time_period has an F1-score of only 0.29. These results suggest that the current model often struggles with entities requiring nuanced understanding or contextual information.

Among these mispredicted labels, "rooms" stands out as particularly problematic. For B-rooms, the F1-score is 0.35, with precision at 0.38 and recall at 0.33. Similarly, for I-rooms, while precision is 1.00, recall is 0.17, resulting in a very low F1-score of 0.29. At the slot-label level, "rooms" exhibits a precision of 0.25, recall of 0.22, and F1-score of 0.24, further emphasizing its frequent misclassification. Despite having a support count of 9 for B-rooms and 12 for I-rooms, the model fails to consistently identify spans related to "rooms" in diverse contexts. This discrepancy between support and performance suggests a critical gap in the model's ability to leverage contextual clues or entity-specific features. As "rooms" is a frequently encountered span in the dataset, addressing these failures is essential for improving overall system performance.

3.2 Hypothesis for Error Sources

The errors in predicting "rooms" are hypothesized to arise from two main factors: insufficient contextual information and the limitations of word embeddings.

First, the lack of contextual information prevents the model from leveraging adjacent tokens (e.g., numbers or descriptive terms) that often indicate the presence of "rooms." For instance, in the example "I want to change my reservation to 3 adults in 2 rooms please," the number "2" directly relates to "rooms." However, the current system does not effectively associate such contextual clues, leading to errors. Without incorporating neighboring tokens, the model cannot discern patterns like the co-occurrence of numbers and room-related terms.

Second, word embeddings alone fail to differentiate "rooms" from semantically similar nouns like "tables" or "spaces." For example, the phrases "book a table for tonight" and "book a room for tonight" might have similar embeddings, causing confusion. Additionally, cases like "I need a single room for 2 adults" involve subtle morphological differences that embeddings alone cannot capture. Using linguistic features like lemma and dependency relations can mitigate this ambiguity by providing more structured representations.

3.3 Linguistic Features Added

To address these limitations, several linguistic features were integrated into the model:

1. **Morphological Features**: These capture fine-grained grammatical details such as number and degree. For instance, in “I need a wheelchair-friendly room,” "room" has the morphological feature Number=Sing, which helps identify it as part of a specific span. Similarly, "friendly" has the feature Degree=Pos, distinguishing it as an adjective modifying "room."

2. **Head Token Information**: By considering the syntactic head of each token, the model gains insight into broader sentence structures. For example, in "I am still waiting for

the refund of the room I canceled," the head token of "room" might provide a stronger indication of its relationship to the overall sentence.

3. **POS, DEP, and Lemma Encoding**: Part-of-speech tags, dependency labels, and lemmatized forms offer additional layers of abstraction. These features help differentiate semantically similar nouns by their syntactic roles. As discussed in Section 3.2, "table" and "room" may have similar embeddings but differ in their dependency labels (e.g., nummod for "room" in "3 rooms").

3.4 Model Enhancements and Evaluation

Initial Enhanced Model: The first enhanced model incorporated the linguistic features mentioned above. Specifically, it included: word embeddings, morphological features for the current token, POS, DEP, and lemma encoding for the current token, its head token, and the immediate previous and next tokens and numeric indicators for neighboring tokens (e.g., whether a token is a number).

This model demonstrated significant improvements in both the accuracy of the "rooms" label and overall performance, which is shown as Table 2. For BIO tags, the F1-score for B-rooms improved from 0.35 to 0.62. Precision for B-rooms increased from 0.38 to 0.71, and recall rose from 0.33 to 0.56. Similarly, the F1-score for I-rooms increased from 0.29 to 0.53, reflecting more consistent tagging within room-related spans. At the slot level, the rooms span label improved significantly. Precision increased from 0.25 to 0.57, and recall rose from 0.22 to 0.44, resulting in an F1-score improvement from 0.24 to 0.50. For overall performance, weighted F1-score for BIO tags increased from 0.52 to 0.61 and weighted F1-score for slot labels rose from 0.46 to 0.56.

Table 2: Comparison between logistic regression model and our model

BIO/SLOT	**Logistic Regression**			**Our Model**			**Support**
	Precision	Recall	F1-score	Precision	Recall	F1-score	
B-rooms	0.38	0.33	0.35	0.71	0.56	0.62	9
I-rooms	1.00	0.17	0.29	0.71	0.42	0.53	12
BIO-weighted-avg	0.62	0.49	0.52	0.70	0.57	0.61	166
slot-rooms	0.25	0.22	0.24	0.57	0.44	0.50	9
slot-weighted-avg	0.46	0.47	0.46	0.57	0.56	0.56	93

Further Exploration: The subsequent model extended the context window to include two tokens on either side. To reduce noise, it simplified the features for neighboring tokens, retaining only their word embeddings and numeric indicators. While this approach improved overall system performance (weighted precision, recall and F1 increased to 0.74, 0.59, 0.64 respectively), the three metrics for "rooms" were slightly lower than the first enhanced model.

This decline indicates that while a broader context window captures additional dependencies, it may introduce noise, particularly for spans like rooms, which rely on local syntactic and semantic cues. For example, in sentences like "Can I book 2 rooms for tonight?" the simpler contextual features of the second model struggled to distinguish rooms from other entities when adjacent tokens had ambiguous roles (e.g., "2" could refer to quantity or time). Therefore, future work could explore dynamic context windows or attention mechanisms to balance the trade-off between context and noise.

Additionally, we also used an example to visually compare the prediction results of the Logistic Regression model and our models, in the sentence "From the 16th to the 30th, a single room for 2 adults and 1 child," the logistic regression model failed to predict "a single room" correctly. Both enhanced models succeeded, demonstrating their ability to capture contextual dependencies.

4 Final Test Reporting and Reflection

4.1 Comparison On Validation Set & Test Set

The results on the validation and test sets are generally in line with our expectations. For the validation set, as previously discussed, our model demonstrated satisfactory performance. The overall metrics were consistently better than those of the logistic regression model. For instance, both BIO tags and slot labels achieving an approximate 10% increase. This indicates that the feature engineering we implemented has been beneficial for the prediction of both BIO tags and slot labels.

For the test set, the classification report shows that the improved model achieves significant improvements in micro-average and weighted-average metrics for both BIO tags and slot labels, particularly in recall and F1 scores, outperforming the logistic regression model by about 10% in these metrics (as shown in Table 3). However, the macro-average metrics saw limited improvement, indicating that certain tags may still underperform. Notably, labels such as ‘B-adults‘, ‘B-date_from‘, ‘B-rooms‘, and ‘B-date_to‘ show substantial F1-score improvements. For instance, in the sentence "Have you got any availability between 17th May to 21st May, we are three for a four night stay," the logistic regression model misclassifies "17th May" as "O B-date_from," whereas the improved model correctly predicts "B-date_from I-date_from." This highlights the logistic regression model’s struggle with multi-token spans and limited contextual understanding. In contrast, the improved model excels in such cases, probably benefiting from richer features such as dependency parsing, morphology, and contextual embeddings, while maintaining strong performance on simpler tags. Overall, the improved model achieves a better balance between accuracy and contextual understanding.

Table 3: Logistic Regression model and our model on test set

BIO/SLOT	**Logistic Regression**			**Our Model**			**Support**
	Precision	Recall	F1-score	Precision	Recall	F1-score	
BIO-micro-avg	0.58	0.40	0.47	0.69	0.52	0.59	192
BIO-macro-avg	0.49	0.36	0.38	0.49	0.39	0.42	192
BIO-weighted-avg	0.61	0.40	0.44	0.69	0.52	0.57	192
slot-micro-avg	0.37	0.39	0.38	0.53	0.49	0.51	98
slot-macro-avg	0.37	0.41	0.38	0.44	0.42	0.42	98
slot-weighted-avg	0.38	0.39	0.38	0.53	0.49	0.51	99

4.2 Reflection

The improved model demonstrates significant advancements in recall and F1 scores for most tags, validating the effectiveness of the added features and refinements, particularly for complex or multi-token spans. However, challenges remain for less frequent or semantically similar tags, where the model exhibits occasional confusion or inconsistency. To address these issues and build on the current improvements, we recommend balancing the training data to enhance the performance of underperforming tags through techniques such as oversampling or data augmentation. Additionally, implementing sequence-level optimization algorithms, such as the Viterbi algorithm, could improve tagging consistency by considering global probabilities and contextual dependencies. These steps would ensure a more balanced and robust model while maintaining the observed improvements for frequently occurring or complex spans.