ANLP Assignment 2 2022
Solution Notes and Marking Information

General Points

A key theme of this assignment was that it is important to look at your data, develop your own hypothesis about what is happening, test it, and convince someone that your results are believable, writing as concisely as possible in your own words. These are critical skills that you will need to demonstrate in any real world application of your NLP knowledge. With that in mind, we noticed a few common problems with some submissions:

- The answers to some questions contained little evidence the data had been analysed. Some contained no examples at all. Some supplied examples, but merely observed that they were correct or incorrect, or that they reflected improvements on some measure (such as F-score). What we were looking for here was for you to *explain* how the examples illustrated some property of a particular model.
- A common pattern that we saw was a tendency to recite known information, such as the formulas for precision, recall, or F1, or to simply report their values (which we already knew for questions 1 and 2). While we didn't penalize anyone for this, we also didn't reward any points for it. On these questions, we were explicitly looking for you to *synthesize* different pieces of knowledge that you've learned, and use them to *analyse* new situations. Reciting known information doesn't demonstrate that you can do this. We are much less interested in the fact that you know how these measures are computed (which is something you can look up in a textbook), than we are in whether you understand when it is appropriate to use them and what their limitations are (which you need to think about in deciding whether to apply them to a particular situation).

A less common problem which still afflicted a handful of submissions was ignoring page limits, or trying to subvert them (e.g. by changing margins). As we clearly warned in the instructions, these submissions were penalised.

Question 1

The main thing that we were looking for was for you to demonstrate that you had compared some example outputs of each model, produced simple hypothesis about why the two models behave

differently on these examples, and explained your hypothesis with the help of the examples. If you did all of these things well, then you will have gotten most of the marks for this question. Most students did well on this question, but we noticed a few failure modes:

- Focusing on metrics to exclusion of examples altogether.
- Presenting examples without connecting them to a hypothesis.
- Talking in general about how the methods work with little to no reference to examples.

Some of you correctly noticed that the word embeddings supplied by the spacy model are *contextual* embeddings, meaning that the embedding for each token is a function of both the word and its surrounding context. One way to see this is that even tokens of the same type will have slightly different embeddings, in some cases producing different predictions (something that a model conditioned on static embeddings cannot do). It is possible to demonstrate this directly using examples, and a few especially observant submissions did so. However, we did see some incorrect claims that logistic regression inherently used context. It does not. Logistic regression can only use the features it is given. For it to use context, the input features must encode context.

Question 2

Most of you correctly identified that the provided algorithm was greedy, and that the Viterbi algorithm could be used for this problem. For high marks, it was also important to:

- Correctly explain what quantities you would substitute for the emission and transition probabilities that are used when applying Viterbi to an HMM. The model we gave you was not an HMM, and it was important that your adaptation did not change its interpretation. (The HMM is a *generative* model of tags and words, while the model we gave you is a *discriminative* model of tags given words.) However, the Viterbi algorithm is simply a way to choose an optimal tagging in a model that includes only local transition and emission quantities; it doesn't require that model to be an HMM, and it works well here. Commonly suggested adaptations that weren't appropriate included:
  - Learning any kind of transition distribution over pairs of tags, or
  - Assigning transition probabilities from each tag to all legal successor tags so that the transition probabilities sum to one. Besides being incorrect as a generative model (since it generates each tag twice), this solution changes the probabilities of the original model non-monotonically — specifically, it boosts the probability of `O` tags. To see why this is so, consider the number of possible transitions from an `O` tag, compared to the number of transitions from an `B-` or `I-` tag.
- Explain how to deal with the beginning of the sequence.
- Analyse the tradeoffs of greedy and Viterbi, in particular with respect to optimality and complexity. The best answers to this question analysed complexity formally.

Some of you noticed that the provided code does not work directly with probabilities, and instead works with log-probabilities. Some very good answers also considered how to deal with this fact.

Question 3

To get high marks on this question, you needed to do a good job on each of the main subtasks:

1. Analysing the data to identify at least one interesting error pattern,
2. Motivating a new feature or features that might improve the model on the error(s) that you identified,
3. Designing and implementing your feature, and
4. Analysing the effect of your new model, both on the target error class that you identified, and the system as a whole; reflecting on why it did or did not work as you hypothesized.

Note that you did *not* need to improve the model in order to get high marks here!

There were many approaches that people took on this question, including:

- using features from neighbouring words, based on the observation that a word's tag can often be disambiguated by using information from its neighbours. For those who did this, it was important to think carefully about how to handle words at the beginning or end of a sentence. If using word embeddings of neighbouring words as features, it was important to think about how to include them: concatenating the vectors is a simple and effective choice; we sometimes saw other, more complicated schemes that were less well-motivated.
- using features from a word's syntactic parent, based on the observation that the parent word can often help disambiguate a word. Here, you would need to decide how to handle a word with no parent.
- Using other features from the dialogue system, such as its intent, based on the observation the certain span types tend to be associated with certain intents. It would not have made sense to use the slot's value (which is determined later in the pipeline that we described to you).

We also saw many other approaches. We emphasize again that it did not matter *which* approach you took, but rather how clearly you motivated it, designed it, and analysed its result.

Question 4

There was not a prescribed answer for this, because the results on the test set were highly dependent on how you approached question 3. Some common problems that we saw:

- We frequently saw statements that speculated on the cause of differences in the behaviour of the model between the validation and test data. In many cases, the speculation was based on a property that was easy to confirm by measuring some property of the validation and test data (e.g. a difference in the distribution of tag types), but which was not in fact measured. To get full marks on this question, you needed claims backed by evidence, not just speculation.

- We frequently saw statements to the effect that test set performance was expected to be worse than development set performance, and overfitting was often cited as a cause. It is of course possible that your models are overfit to the training data, or developed with such narrow focus on peculiarities of the validation that data they are overfit to it. But you can't infer this by comparing aggregate scores of a single model on the validation and test data. These aren't directly comparable since the datasets are different, and you would not expect them to behave identically.