Dependency Parsing-Assignment 3

Arkajyoti Pal
15CS30003
arkapal.pal@gmail.com

1 Features and Feature Combinations Considered

Overall, the following features have been considered to be taken in consideration for feature combinations:

- **ID**: Word index, integer starting at 1 for each new sentence; may be a range for multiword tokens; may be a decimal number for empty nodes.
- **FORM**: Word form or punctuation symbol.
- **LEMMA**: Lemma or stem of word form.
- **UPOS**: Universal part-of-speech tag.
- **XPOS**: Language-specific part-of-speech tag; underscore if not available.
- **FEATS**: List of morphological features from the universal feature inventory or from a defined language-specific extension; underscore if not available.
- **HEAD**: Head of the current word, which is either a value of ID or zero (0).
- **DEPREL**: Universal dependency relation to the HEAD (root iff HEAD = 0) or a defined language-specific subtype of one.
- **MISC**: We also use the additional features provided in the 10th column similar to the *morphological features*.

Specifically, three feature combination schemes have been considered:

- We use all the above mentioned features except the morphological features(*FEATS*) and additional features in the 10th column.
- We use all the above mentioned features except the additional features in the 10th column.
- We use all the above mentioned features including the additional morphological features provided in the 10th column of the data.

1.1 Suppressing and Taking Features

To suppress and take any particular feature, the $extract\_feature$ member function of the *Configuration* class has been modified. Specifically,

- To retrieve the additional features in the 10^{th} column we augment the $extract\_feature$ member function to handle the additional features in the same way the morphological features have been handled.
- To supress features, two global flags, namely *misc_status* and *morph_status*, are used to indicate whether the additional features in the 10^{th} column and/or morphological features should be taken in the $extract\_feature$ member function respectively.

2 Evaluating Performance

The tuple in the table entries represent the labelled and unlabelled score. The feature combination schemes are :

- **Scheme-I**:We use all the features mentioned in the Features' section except the morphological features(*FEATS*) and additional features in the 10th column.
- **Scheme-II**:We use all the features mentioned in the Features' section except the additional features in the 10th column.

- **Scheme-III**:We use all the features mentioned in the Features' section including the additional morphological features provided in the 10th column of the data.

Table 1: Performance of **Arc-Standard** Transition Parsing for the various machine learning models and Feature combinations

	Logistic Regression	**SVM**	**MLP**
Scheme-I	(0.80, 0.69)	(0.85, 0.77)	(0.82, 0.71)
Scheme-II	(0.80, 0.69)	(0.86, 0.77)	(0.82, 0.71)
Scheme-III	(0.87, 0.78)	(0.86, 0.77)	(0.90, 0.80)

Table 2: Performance of **Arc-Eager** Transition Parsing for the various machine learning models and Feature combinations

	Logistic Regression	**SVM**	**MLP**
Scheme-I	(0.85, 0.74)	(0.87, 0.77)	(0.85, 0.74)
Scheme-II	(0.86, 0.75)	(0.88, 0.79)	(0.88, 0.77)
Scheme-III	(0.90, 0.80)	**(0.91, 0.83)**	(0.90, 0.81)

3 Conclusion

As we can see from Table 1 and 2, 'Arc-eager' transition parsing performs consistently and considerably better than the 'Arc-standard' transition parsing across the machine learning models and feature combinations. Also, SVM gives the best performance among the machine learning models with the 'Arc-eager' transition parser. Also, the comparatively poor performance of neural network based classifier can be attributed to the small amount of training data available.