AUX vs. VERB

Observations Pertaining to the Problem Statement

According to the definition in UD[1], `AUX` is used as a common POS tag for verbal auxiliaries, as well as non-verbal TAME markers. The class of copulas are also included in this list.

This definition of auxiliaries is a bit different from Shopen [2007] which separates the two classes of auxiliaries and copulas in different categories. The work also points out the correlation between the position of an inflected auxiliary in relation to the verb, and other word properties of the language, as first pointed by Greenberg [1963]. In his work, Greenberg notes that the position of an inflected auxiliary in relation to the verb is generally the same as the position of verb in relation to an object. It is important to note that this generalization only holds for the inflected auxiliaries, and thus languages where the auxiliaries are not inflected are automatically ruled out from the consideration. Shopen points out the well-known exception to this generalization in case of verb-second languages like those of German.

While the generalization made by Greenberg is a very good marker for possible identification of inflected auxiliaries, the requirement of identification of auxiliaries in noninflected form still remains as a problem. This problem can however, be mitigated in part by the usage of the list of tokens identified as auxiliary in a given language, as was started in UDv2.4 with the help of a validator (cf. Level 5 checks in `validate.py`[2] file). It must also be pointed out that since Greenberg did not extend this generalization to SVO languages, the generalization only holds for languages with VSO and SOV dominant word-order languages. Combining that with verb-second languages, the generalization can not be used globally across all the languages.

When the copulas are included in the definition of `AUX`, the already difficult problem of separating `AUX` and `VERB` becomes even harder. In many languages, auxiliaries are a subset of verbs, with respect to specific usages. In other words, the same token can act as a verb or an auxiliary, depending upon the usage. The list of copula in many languages is also a subset of verbs, called as copulative verbs. However, as Shopen notes, there are cases of languages where the copula are not verbal in nature. The function of a copula can be realized by other means as well. The most common of these, viz. juxtaposition (example language- Ilocano), and use of predicators (example language- `bm`) are listed in the work, where they may be combined with existing copulative verbs in the grammar of the language.

In essence, while the class `AUX` in UD includes the copulative verbs, predicators, and other non-verbal TAME markers, the class `VERB` is composed of open class categories of verbs.

[1] `https://universaldependencies.org/kpv/pos/AUX_.html`

[2] `https://github.com/UniversalDependencies/tools`

Dataset Definition

This experiment was initially tried on UDv2.3, but failed terribly. With the release of UDv2.4, this experiment was tried again, keeping the dataset treebank same, but changing the model architecture et al. In the current documentation, we will use `hi`-hdtb treebank from UDv2.4.

There are a few reasons for the choice of the language for the experiment. In `hi`, we can more often than not draw a clear line of distinction between auxiliary as defined by UD, and the verbs. While the auxiliaries undergo inflection, and also include predicators and other TAME markers, they are restricted to a few tokens which rarely, if at all, are used as independent verbs. The factors as listed above, combined with the author's native fluency in the language makes it an ideal candidate for this experiment.

Experiment

We approach the problem at hand as a classification problem, specifically as a Sequence Labelling based NER problem. As part of this measure, we convert the data from the entire `hi` data to a format that suits the task[3].

There exist two tag formats for NER, namely IOB and IOBES. While IOB is composed of 3 tags- Inside, Outside, Begin; the IOBES tagset extends the IOB tagset by adding End and Singleton tags. The IOBES tagset helps with the better annotation of the data, as it provides more information. For example, in IOB tagset, the singleton entities are labelled with 'B' tag, without any following elements covered by 'I' tag. In IOBES, the tag 'S' is used to specify a single element being tagged. Similarly, the end of a sequence is not marked explicitly by IOB tagset, but is done with 'E' tag in IOBES format.

To convert our data into IOBES format, we use the following methodology. All the instances marked as `AUX` are labelled as "S-aux", and all the instances marked as `VERB` are labelled as "S-verb". The rest of the tokens are labelled with 'O' tag. We do not consider contiguous tokens as a continuous chain, and thus not use either of 'I', 'B' or 'E' tags at all. This is also done so as to have better control over each token that the model learns to tag, thereby increasing the granularity of the data.

For the task of NER, as well as POS Tagging, Flair embeddings [Akbik et al., 2018] were the SOTA at the time of performing this experiment. The embeddings were shown to be outperform several models available at the time, across multiple NLP tasks, and therefore were the natural choice for this experiment. However, there are several hyper-parameters that can be tuned with respect to the models. We decided to tune the hyper-parameters with their corresponding choices as listed in Table 1. The best choice for the hyper-parameter are also listed in the same table.

Hyper-Parameter	**Choices**	**Tuned Value**
Embeddings	**Stack1:** Forward and Backward Flair Embedding trained on `hi`-newswire	Stack2
Continued on next page		

[3]Code available at `https://github.com/Akshayanti/aux_verb.git`

Hyper-Parameter	Choices	Tuned Value
	Stack2: Word Embedding for `hi`, Forward and Backward Flair Embedding trained on `hi`-newswire	
Use CRF?	True, False	True
Use RNN?	True, False	True
RNN Layers	1, 2, 4	2
Size of Hidden Layer	32, 64, 128, 256	256
Dropout	Uniform Distribution in [0.0, 0.5]	0.25
Learning Rate	0.05, 0.1, 0.15, 0.2, 0.25	0.1

Table 1: Hyper-Parameters for Neural Network

Since we are trying to correct the gold standard itself, we are very liable to run into a cold start problem. To counter this problem, we perform a k-fold cross-validation on the data, with k=10. We concatenate the different splits of the treebank (train, dev, test) and then split the data into 10 folds, with test set in each fold disjoint with other test data in other folds. We then proceed to convert each of those folds into the IOBES tagset as we did with the unmodified data.

With the optimized parameters, we train models on each fold of the data. The trained model instances are then used to predict the test set in each fold as well. The output of the evaluation writes the predictions with the associated confidence in the predicted label. We here identify the 6 kind of patterns as listed in Table 2.

Category	Original	Prediction
aux_TP	S-aux	S-aux
O_TP	O	O
verb_TP	S-verb	S-verb
aux-O	O	S-aux
	S-aux	O
aux-verb	S-aux	S-verb
	S-verb	S-aux
verb-O	O	S-verb
	S-verb	O

Table 2: Categories of Error Patterns

Since we also have confidence scores associated with each prediction, we focus on a set of error patterns within certain bounds on the confidence scores. Figure 1 shows the distribution of confidence scores for instances where the predicted label matches the original label, with the associated confidence value lower than 0.80. For these categories, we focus on the subset where the confidence score is lower than 0.67. The idea is that since there are 3 categories, a prediction with a confidence lower than $\frac{2}{3}$ is liable to be erroneous. For the instances where there is a mismatch between the predicted label and the originally annotated label, we focus on instances with the confidence in prediction higher than 0.995. The idea in this case is that if the model is really sure about the prediction, the original

annotation might be erroneous, and is worth looking into.

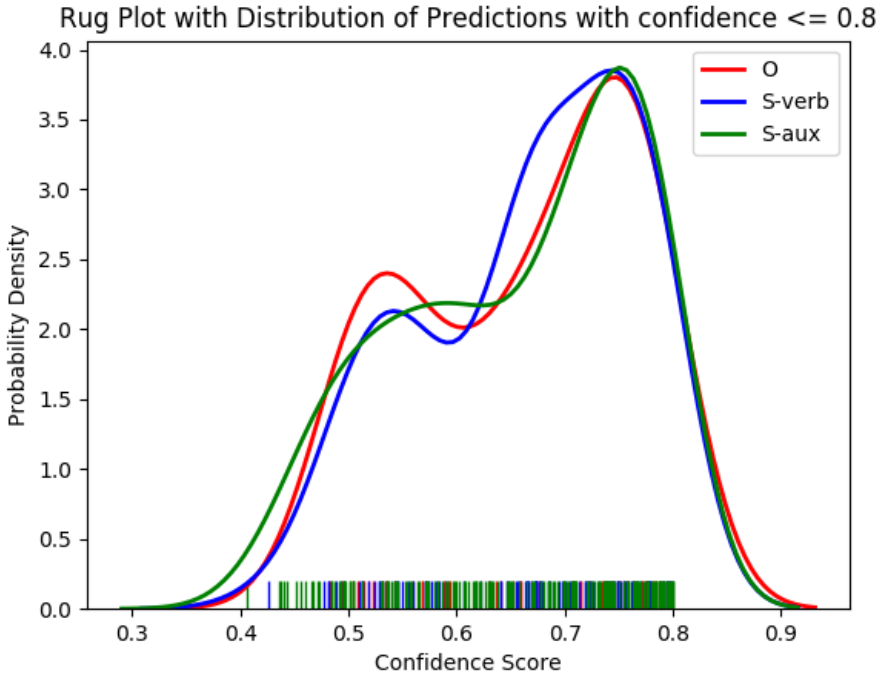

Figure 1: Rug plot with Distribution of Predictions with low confidence score

Having identified instances within each category that have confidence scores within the relevant bound, these instances were manually annotated to see if they were actually mislabelled patterns or not. We can summarize the entire experiment in the form of algorithm as defined in Algorithm 1.

Algorithm 1 Experiment to Identify Mislabelled `AUX` and `VERB` tags

Input: *data* ← UDv2.4 treebank

Convert *data.train*, *data.test* and *data.dev* to IOBES format
Optimize Sequence Labelling NER model configurations for the *data*
$model.config$ ← best sequence labelling NER model configuration
$data.complete \leftarrow data.train + data.dev + data.test$
{The different splits of the data concatenated together}
iter.id ← fold of *data.complete*, numbered as *id*
{Performed 10-fold cross-validation to split *data.complete*}
model ← NER model with $model.config$ configuration
for *id* in {1, ... , 10} **do**
    model.id ← *model* trained on *iter.id.train* data
    model.id.test ← Prediction of *model.id* on *iter.id.test* data
end for
$identified.pure$ ← Instances identified as True Positive across all *model.id.test*
{Confidence score ≤ 0.6700}
$identified.cross$ ← Instances identified as False Positive or False Negative across all *model.id.test*
{Confidence score ≥ 0.9950}
Manual Annotation of $identified.pure$ and $identified.cross$

Results

The output of running the NER tagger on test data within each of the fold returns the predictions of the model, and an associated confidence score value. Also, owing to multi-class classification, the model performance is expressed in form of confusion metrics for each class `AUX`, `VERB` along with the metrics like Precision, Recall, Accuracy, F1 Score.

The metrics corresponding to the best performing model on the original treebank is listed in Table 3. When the models were trained on each of the folds, keeping the architecture of the best model, there was no loss in performance of the trained models (metric considered- micro averaged F1 score).

Label	Precision	Recall	Accuracy	F1 Score
`AUX`	98.89	99.50	98.40	99.19
`VERB`	99.32	98.87	98.20	99.09

Averaging	Accuracy	F1 Score
Micro	98.29	99.14
Macro	98.30	99.14

Table 3: Metrics of Best Model trained over original `hi` data

As mentioned in previous section, we focused on the instances of the tagged data with confidence scores in particular bounds. Table 4 lists the number of instances that were focused on in each category (as defined in Table 2). The table

also lists the number of instances that were identified as mislabelled, following the annotation procedure.

Category	Focused	Mislabelled	Percentage
aux_TP	83	3	3.61
O_TP	25	5	20.00
verb_TP	45	10	22.22
aux-O	10	9	90.00
aux-verb	42	23	54.76
verb-O	20	11	55.00
Overall	225	61	27.11

Table 4: Results of Manual Annotation

Discussion of the Results

Metric	Count
Sentences	16 647
Words	351 704
Tagged `AUX`	26 030
Tagged `VERB`	33 753

Table 5: Statistics for `hi` data

Table 5 lists the counts of sentences and the number of `AUX` and `VERB` tags in the entire `hi`-hdtb treebank. Of the total number of tags listed in either category, we are able to focus on just 225 instances where we might be able to identify the problems. Even out of those 225, the success ratio is less than 30%. While certain patterns are more reliable than others (the case where predicted labels don't match the annotated labels), the numbers are not significant enough for the process to be automated.

Acknowledgements

Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures"

Computational resources were supplied by the Ministry of Education, Youth and Sports of the Czech Republic under the Projects CESNET (Project No. LM2015042) and CERIT-Scientific Cloud (Project No. LM2015085) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

Bibliography

Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

Joseph H Greenberg. Some universals of grammar with particular reference to the order of meaningful elements. *Universals of language*, 2:73–113, 1963.

Timothy Shopen. *Language Typology and Syntactic Description*, volume 1, pages 40–59. Cambridge University Press, 2 edition, 2007. ISBN 0-511-36671-X. doi: 10.1017/CBO9780511619427.