A Neural Network Approach to Ellipsis Detection in Ancient Greek

Giuseppe G. A. Celano
Leipzig University
Faculty of Mathematics and Computer Science
Institute of Computer Science
celano@informatik.uni-leipzig.de

Abstract

In the present article, five neural networks models for prediction of the number of elliptical nodes in Ancient Greek sentences are compared. The models are trained on dependency treebank data, where elliptical nodes are introduced if and only if they govern nodes that would otherwise become orphans. As exact word forms of elliptical nodes cannot often be identified (and therefore be annotated) in Ancient Greek, the task is modeled as a multiclass classification one, where each sentence is associated with zero, one, two, or more than two elliptical nodes. The study shows that pretrained BERT token embeddings allow achievement of the best performance. A model, which is the first of its kind, is made available for further research.

1 Introduction

In linguistics, "ellipsis" can be broadly defined as the phenomenon whereby a sentence lacks one or more constituents that are left implied, but can be inferred from the linguistic context.

Depending on the language analyzed and the theoretical framework, different descriptions and definitions of ellipsis have been proposed in the theoretical linguistics literature (see, for example, Van Craenenbroeck and Temmerman, 2019 for a general overview).

Ellipsis in Ancient Greek has often been associated with, or treated as, a stylistic device in the older literature (see, for example, Kühner et al., 1965, pp. 558–571, who provide a long list of examples, and Schwyzer, 1971, pp. 707–710). In the more recent literature, however, the phenomenon has been investigated in word order studies. Gaeta and Luraghi (2001), for example, distinguish three different types of ellipsis: gapping, split coordination, and coordination reduction. Gapping occurs when there are at least two contrasted constituents:

(1) ὥσπερ Ἐμπεδοκλῆς φησὶ$_i$ φιλίαν, ἄλλος δέ τις Ø$_i$ πῦρ, ὁ δὲ Ø$_i$ ὕδωρ ἢ ἀέρα
'as Empedocles holds of Love, another thinker of fire, and another of water or air'[1] (Arist. Metaph. 996a 8)

In Example (1), the contrasted constituents are, on the one hand, "Ἐμπεδοκλῆς," "ἄλλος (δέ) τις," and "ὁ (δὲ)," and, on the other, "φιλίαν," "πῦρ," and "ὕδωρ ἢ ἀέρα," the elliptical constituent being the verb "φησὶ." This sentence is an example of rightward gapping, because the elliptical verbs refer back to "φησὶ." This is not the only type of gapping in Ancient Greek, in that an elliptical constituent could also refer to a following constituent (leftward gapping).

Examples of split coordination and coordination reduction are given by Examples (2) and (3) , respectively:

(2) ἕπεσθαι$_i$ δέ οἱ$_z$ τῶν μαχίμων μὲν οὐδένα ἀνδρῶν, καπήλους δὲ καὶ χειρώνακτας καὶ ἀγοραίους ἀνθρώπους Ø$_{iz}$
'and none of the warriors would go with him, but only merchants and craftsmen and traders' (Hdt. 2.141.4)

(3) ἱστία$_i$ μὲν στείλαντο, θέσαν δ᾽ Ø$_i$ ἐν νηῒ μελαίνῃ
'they furled the sail, and stowed it in the black ship' (Hom. Il. 1.433)

In Example (2), the verb and its second argument "οἱ" are omitted, while in Example (3) only the object is.[2] Ellipsis of direct object (and some other second arguments)[3] often occurs also in complex

[1]Translations in the article derive from the Perseus Digital Library at http://www.perseus.tufts.edu/hopper/.

[2]As Ancient Greek has a rather free, information structure-based word order, indication of the position of the ellipsis in the examples is to be considered approximate.

[3]I leave aside the question of the relationship between ellipsis and verb valency in Ancient Greek, which has not yet been investigated satisfactorily. Indeed, in some examples,

sentences, as in Example (4) (Luraghi, 2003, p. 170) :

(4) ὁ δὲ ἐμπιμπλὰς ἁπάντων$_i$ τὴν γνώμην, ἀπέπεμπε Ø$_i$
'having satisfied the expectation of all, he dismissed them' (Xen. Anab. 1.7.8)

The variety of ellipsis types, of which Example (1)–(4) just offer a meager glimpse, is clearly dependent on information structure, which has been proved to determine the high configurational complexity of Ancient Greek word order (Celano, 2013; Dick, 1995). As can be expected, therefore, it is not only challenging to describe and explain ellipsis in Ancient Greek, but also to annotate it.

In the following sections, I present five models built to predict the number of elliptical nodes in Ancient Greek sentences. More precisely, an overview of related work is given in Section 2. Section 3 provides details on the data used for the present study, while Section 4 describes ellipsis annotation. Five models to predict ellipsis are compared in Section 5. A few concluding remarks are contained in Section 6.

2 Related Work

Most of the previous ellipsis-related work conducted in computational linguistics/NLP has so far focused on verb phrase ellipsis (VPE) in English.[4]

Hardt (1997) describes a rule-based system to resolve ellipsis in 644 examples from the Penn Treebank and, more recently, Hardt (2023) tests the ability of a number of Large Language Models to understand ellipsis. Nielsen (2004) tests a variety of machine learning algorithms for VPE detection using the British National Corpus and the Penn Treebank. Bos and Spenader (2011) provide an account for the creation of a new VPE corpus, by detailing the annotation process of the 25 sections of the Wall Street Journal contained in the Penn Treebank. Bos and Spenader's (2011) corpus has also been used by Zhang et al. (2019), which seems to be the first neural networks-based study on VPE.

In the above-mentioned literature, VPE processing for English is divided into two main related tasks: (i) VPE detection and (ii) VPE resolution.

one might posit one-argument verbs instead of two-argument verbs with ellipsis of an object.

[4]Noun ellipsis detection, which is less relevant for the present study, has recently been investigated by Khullar (2020).

Codepoint	F	RF
Greek Coronis (U+1FBD)	7,224	0.16
Combining Comma Above (U+0313)	28,581	0.63
Apostrophe (U+0027)	11	0.00
Right Single Quotation Mark (U+2019)	4,481	0.10
Modifier Letter Apostrophe (U+02BC)	5,269	0.12
	45,566	1

Table 1: Unicode characters used to encode the apostrophe in the original treebank data with their (relative) frequencies.

VPE detection is modeled as a binary classification task outputting whether or not auxiliaries, such as "do," "be," or "have," are used as triggers, i.e., they replace a preceding VP. On the other hand, VPE resolution aims to identify the antecedent a given trigger refers to: more precisely, the task involves identification of candidate antecedents, over each of which a binary classification task is performed (Zhang et al., 2019).

As will be shown in the following sections, the task at hand to predict ellipsis in Ancient Greek differs from the above-mentioned studies because it only concerns detection of the number of elliptical nodes in a sentence, without identification of its word form or resolution. This task indeed depends on the nature of the Ancient Greek language, where, typically, there is no trigger constituent for an elliptical node, and its exact form and position are often unclear.

3 The Data

There exist two major related data sets containing morphosyntactic annotations of Ancient Greek texts, which also include annotation for ellipsis: the Ancient Greek Dependency Treebank[5] and the Dependency Treebanks of Ancient Greek Prose (Gorman, 2020).[6] [7]

The two treebanks, which have been annotated using the same annotation scheme, have been merged together (for convenience, I henceforth refer to this data set as "Ancient Greek Dependency Treebank"): the data set comprises 187 files,

[5]`https://github.com/PerseusDL/treebank_data/releases/tag/v2.1_IGDS`.

[6]I downloaded the data from the main branch of `https://github.com/vgorman1/Greek-Dependency-Trees`, which contains more recent data than the released one at `https://zenodo.org/record/3596076#.XlZ7CxP7Su4`.

[7]I limited the study to the above-mentioned data sets. There exist, however, a few others: in particular, Pedalion, which is available in a beta version, is worthy of note (Keersmaekers et al., 2019).

Model	Class Weights	Accuracy	Precision		Recall		F1	
			M	w	M	w	M	w
Feedforward$_{Baseline}$		0.77	0.42	0.70	0.27	0.77	0.27	0.71
	✓	0.67***	0.33	0.70	0.34	0.67	0.33	0.68
Feedforward$_{DBBE-BERT}$		**0.85*** **	**0.64**	**0.84**	0.50	**0.85**	**0.55**	**0.84**
	✓	0.80***	0.58	0.81	**0.53**	0.80	0.52	0.80
Transformer$_{DBBE-BERT}$		**0.85*** **	0.61	0.83	0.50	**0.85**	0.54	**0.84**
	✓	0.79***	0.54	0.81	**0.53**	0.79	0.51	0.79
Transformer$_{WordPiece35000}$		0.78**	0.44	0.71	0.28	0.78	0.29	0.71
	✓	0.54***	0.31	0.69	0.39	0.54	0.26	0.59
LSTM$_{DBBE-BERT}$		**0.85*** **	0.61	0.83	0.49	**0.85**	0.53	0.83
	✓	0.81***	0.57	0.81	**0.53**	0.81	0.53	0.81

Table 2: Model metrics calculated on the test set (M = macro, w = weighted). Statistical differences from Feedforward$_{Baseline}$ are calculated using Stuart-Maxwell tests and reported in the accuracy column ($p < 0.05$ (*), $p < 0.01$ (**), and $p < 0.001$ (***)). The highest values are indicated in bold.

amounting to 54,925 sentences and 1,063,984 tokens.[8] In order to facilitate further processing, the data set has been normalized with respect to (i) Unicode form and characters and (ii) tokenization scheme.

The texts have been NFC normalized, and there has been an attempt to make the encoding of the apostrophe uniform. There are at least five different Unicode characters with indistinguishable glyphs that are interchangeably used in the treebank texts (see Table 1): they have all been converted into Modifier Letter Apostrophe (U+02BC).

There has also been an attempt to normalize the data with respect to the tokenization scheme: indeed, while most Ancient Greek graphic words coincide with morphosyntactic words (and therefore with the tokens found in the Ancient Greek Dependency Treebank), there are two main cases where this does not hold true: (i) (negative) conjunctions and (ii) words contracted by crasis.

Conjunctions include examples such as "οὐδὲ" ("and/but not"), which is split into "οὐ" ("not") and "δὲ" ("and/but"): there are 12 such conjunctions[9] and, in the final data set, they have all been segmented.

The words contracted by crasis are words that are univerbated for phonological reasons: an example is "κἀγὼ," which consists of the word "καὶ" ("and") and "ἐγὼ" ("I"). In the original treebank data, about half of all crases are split: it has been found heuristically that 1,371 cases of crasis are split, while 1,110 are not.[10] Since identification, segmentation, and morphosyntactic analysis of crases is challenging, they have not been modified in the final data set. The data and the best model (Feedforward-DBBE-BERT) are made available online[11] for further research.

4 Ellipsis annotation

There are two main challenges with reference to ellipsis annotation in treebanks: (i) identification of ellipsis and (ii) its formal representation.

In Ancient Greek, ellipsis is typically not signaled by a trigger such as auxiliaries in English. On the contrary, its presence can be inferred from linguistic context, as the following example shows:

(5) κρατουμένων μὲν γὰρ ἐπίστασϑε ὅτι πάντα ἀλλότρια
'for when men are conquered, you are aware that all their possessions become the property of others' (Xen. Anab. 3.2.28)

In Example (5), the object clause can be properly annotated only by positing existence of an elliptical verb connecting "πάντα" and "ἀλλότρια," as Figure 1 shows. Indeed, there is only one annotation rule for ellipsis annotation in the Ancient Greek Dependency Treebank: an elliptical node is recog-

[8]The data set is made available at https://git.informatik.uni-leipzig.de/celano/ellipsis_Ancient_Greek.

[9]The full list is available at https://git.informatik.uni-leipzig.de/celano/ancientgreeknlp/-/blob/master/tokenize/texts/to-tokenize.xml.

[10]According to this calculation, crases would amount to about 0.23% of all treebank tokens.

[11]https://git.informatik.uni-leipzig.de/celano/ellipsis_Ancient_Greek.

nized if and only if it is necessary to build a correct syntactic tree, i.e., the posited elliptical node has syntactic dependents.

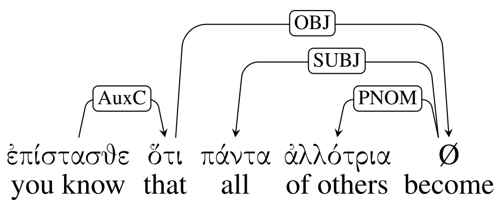
OBJ
SUBJ
AuxC
PNOM
ἐπίστασθε ὅτι πάντα ἀλλότρια Ø
you know that all of others become

Figure 1: (Partial) linguistic tree for Xen. Anab. 3.2.28.

If an elliptical node, as in Example (5), meets this condition, a new token is added at the end of the relevant sentence: such formalization has the advantage of allowing for the elliptical node to function as any other node, and therefore receive annotation for its head and its syntactic function. A disadvantage of this annotation is however that the number of original sentence tokens changes unpredictably, and therefore comparison of different annotations, as well as further machine learning processing, presents an added layer of complexity.[12]

In Example (5), the elliptical verb is likely to be "γίγνομαι," to be added after "ἀλλότρια": however, both form and position of an elliptical constituent are often ambiguous in practice. For this reason, ellipsis representation has been normalized in the data set, so that its form always corresponds, conventionally, to a number in squared brackets (e.g., `[0]`)[13], and its position is always at the end of a sentence, after any other non-elliptical node.

5 Experiment

Prediction of the number of elliptical nodes in Ancient Greek sentences has been modeled as a multiclass classification task, with 4 class labels for (i) none, (ii) 1, (iii) 2, and (iv) 3 or more elliptical nodes per sentence, respectively.

As Figure 2 shows, sentences with no ellipsis and up to 3 elliptical nodes represent about 99.5% of all sentences: for better model performance, therefore, rare sentences with more than 3 elliptical nodes have been included in the class representing 3 elliptical nodes.

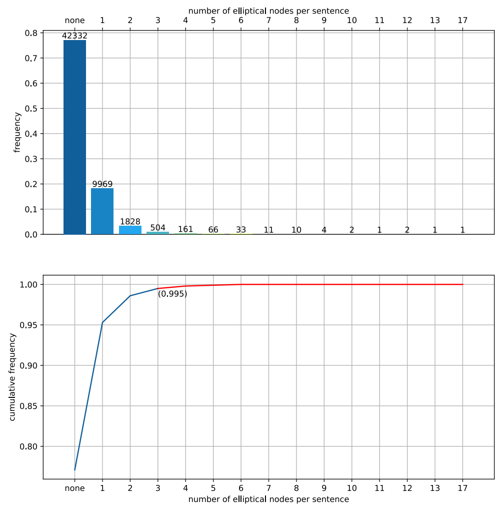

Figure 2: Key statistics for elliptical nodes in the whole data set.

In the following sections, I compare five machine learning models. Each of them has also been trained with class weights[14] because of the class-unbalanced data set. The data set has been divided into training (∼80%), development (∼10%), and test (∼10%) data sets.[15] Accuracy, macro- and weighted-average precision, recall, and F1 are the metrics used for evaluation. To evaluate statistical significance, Stuart-Maxwell marginal homogeneity tests are used.

5.1 Model Architectures

A feedforward neural network has been chosen as a baseline model (Feedforward-Baseline), and its output has been compared to those of four different models: (i) a feedforward model with pretrained BERT token embeddings (Feedforward-DBBE-BERT); (ii) an encoder-only transformer with pretrained BERT token embeddings (Transformer-DBBE-BERT); (iii) an encoder-only transformer with randomly initialized token embeddings (Transformer-WordPiece35000); (iv) an LSTM neural network with pretrained BERT

[12]This is probably the reason why, in Universal Dependencies, ellipsis is annotated at the level of syntactic label (`https://universaldependencies.org/u/overview/specific-syntax.html#ellipsis`): this solution, however, suffers from the disadvantage of rendering syntactic annotation rather obscure.

[13]`[0]` means one elliptical node, `[1]` two elliptical nodes, and so on and so forth.

[14]Weights are calculated using the method `compute_class_weight` of `sklearn.utils.class_weight` (with `balanced` argument).

[15]Because of unbalancedness, the development and test data sets have been selected through stratified sampling, with *strata* being the classes of the dependent variable.

token embeddings (LSTM-DBBE-BERT).[16]

As Figure 3 shows, the baseline model consists of 3 ReLU-activated linear layers (with 5,000, 2,000, and 1,000 units, respectively), a 0.2 dropout layer, and a final linear layer with softmax activation. Sentences are vectorized using TF-IDF scores calculated using the sentences themselves as documents.

An already existing BERT model (DBBE-BERT) (Singh et al., 2021)[17] has been fine-tuned for the Feedforward-DBBE-BERT model. The DBBE-BERT model is a Masked Language Model that has been trained on Modern and Ancient Greek data. The tokenizer vocabulary of the model consists of 35,000 tokens and each token embedding has 768 dimensions. The DBBE-BERT token embeddings have been fed into a global average pooling layer followed by a 0.2 dropout layer, two linear layers with 500 and 100 units respectively, a 0.2 dropout layer, and the final softmax-activated linear layer outputting probabilities (the model architecture is the same as that of the LSTM model shown in Figure 6, except for the LSTM layer, which is replaced by a global average pooling layer).

Since sentences are composed of words, an encoder-only transformer has been tested to leverage relationships between them. Two variants of the encoder-only transformer have been tested.

The first variant (see Figure 4) aims to test whether a transformer layer can improve the performance of the DBBE-BERT token embeddings, which also rely on a transformer-based architecture.

The DBBE-BERT token embeddings are fed into an encoder-only transformer based on Vaswani et al. (2017), mainly consisting of a 2-head attention layer and a feedforward network with two 768-unit linear layers. The output of the transformer is then fed into a further feedforward component, with two linear layers with 500 and 100 units, respectively. All dropout and layer normalization layers have arguments 0.2 (rate) and 1e-6 (epsilon), respectively.

The second variant of the encoder-only transformer aims to assess the contribution of pretrained DBBE-BERT token embeddings. The model architecture is the same as the one described above (see Figure 4), except for token embeddings, which are not pretrained, but randomly initialized.

The tokens for the latter model are identified by a WordPiece algorithm with a vocabulary of 35,000 based on the texts of *Opera Graeca Adnotata* (Celano, 2023), a 34,172,140 token standoff annotation corpus. The texts in *Opera Graeca Adnotata* come from the Perseus Digital Library and First1KGreek projects,[18] and therefore coincide, for the most part, with those used to calculate the DBBE-BERT token embeddings. Each randomly initialized token embedding has 2,000 dimensions, with the transformer encoder's feedforward network consisting of two linear layers with 2,000 (and not 768) units each.

The last tested model is a LSTM neural network (see Figure 6), with a LSTM layer of 1,000 units, followed by two linear layers with 500 and 100 units, respectively, and ReLU activation (the rate of both dropout layers is 0.2). Token embeddings are calculated by the same pretrained DBBE-BERT model used for the Transformer-DBBE-BERT and Feedforward-DBBE-BERT models.

All models have been trained with the Adam optimizer with a learning rate of 1e-6. Early stopping has been determined by monitoring validation loss with a patience of 2 epochs.

5.2 Results

As Table 2 shows, the baseline model without class weights seems to achieve a good accuracy score: however, this score is misleading, in that it is the same as that of a dummy classifier always predicting the most frequent class label (i.e., the `none` label, which means absence of elliptical nodes).

Notably, Transformer-WordPiece35000 without class weights provides results very similar to the baseline ones (see also Figure 5 and 7). The model performance turns out to be statistically different from the baseline's one according to a Stuart-Maxwell test, whose probability value is greater than 0.001, but lower than 0.01. The feedforward model, the transformer, and the LSTM model trained with the DBBE-BERT token embeddings (without class weights) show the best results, with accuracy scores that are 8% higher than the baseline's one.[19]

[16]These token embeddings, as emerges in the following paragraphs, are the DBBE-BERT ones.

[17]The model is called by the authors "Extended Ancient Greek BERT" and, in this paper, "DBBE-BERT" for brevity's sake (DBBE is the acronym of the project "Database of Byzantine Book Epigrams," within which the model was developed).

[18]`https://github.com/PerseusDL/canonical-greekLit`; `https://github.com/OpenGreekAndLatin/First1KGreek`.

[19]The Feedforward-DBBE-BERT model is made available

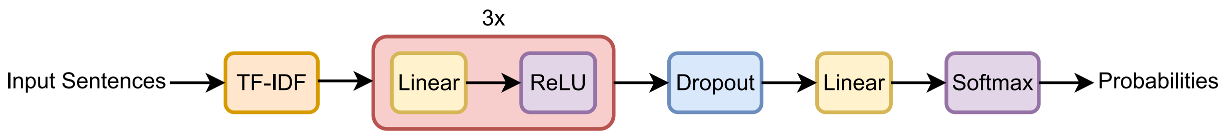

Figure 3: Baseline model architecture.

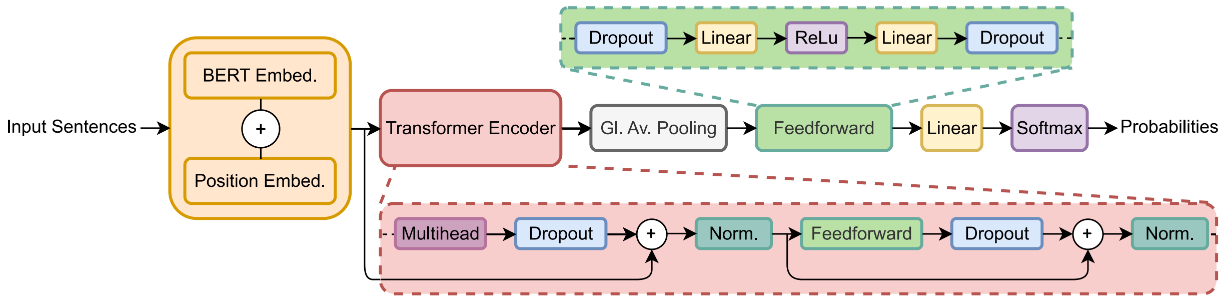

Figure 4: Encoder-only transformer architecture.

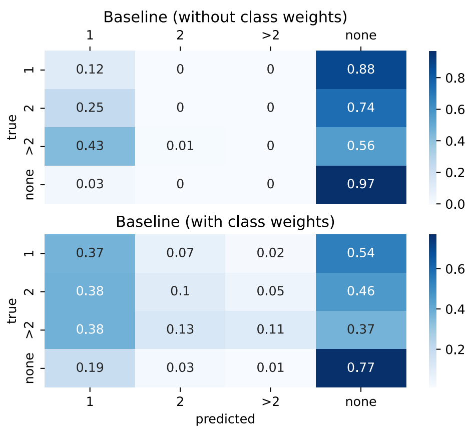

Figure 5: Confusion matrix for the baseline model showing false negatives rates and recall (test data set).

The performances of models with class weights are worse than those of the corresponding models without weigths, and the their differences are statistically significant ($p < 0.001$).

Feedforward-DBBE-BERT's, Transformer-DBBE-BERT's, and LSTM-DBBE-BERT's performances are comparable: however, when their outputs are tested with Stuart-Maxwell tests (pairwise), only Feedforward-DBBE-BERT's and LSTM-DBBE-BERT's results turn out to be not statistically different ($p > 0.05$).

The results suggest that the pretrained DBBE-BERT token embeddings play a crucial role. The transformer architecture of Transformer-WordPiece35000 guarantees a context-aware token representation, but this seems to be not enough for the task at hand, if token embeddings are randomly initialized.

at https://git.informatik.uni-leipzig.de/celano/ellipsis_Ancient_Greek.

5.3 Error Analysis

As Figure 5 shows, the baseline model without class weights can almost exclusively classify sentences as belonging to class none and 1. Most sentences of class none are classified correctly (recall 0.97), even if the classifier also tends to incorrectly label as none most sentences with classes 1, 2, and >2. 12% of the sentences with label 1 are classified correctly, but most of the sentences the classifier labels as 1 are misclassified (see also Table 3). The baseline model with class weights identifies more sentences of class 2 and >2, but precision and recall scores for these classes are very low (0.08, 0.1 and 0.13 and 0.11, respectively), and, more in general, its overall performance, as shown by Table 2, is worse than that of the baseline model without weights.

Figure 7 shows that the confusion matrix for Transformer-WordPiece35000 is surprisingly comparable to the baseline's one (without weights), in that there are almost no predicted sentences of class 2 or >2, and the classifiers' scores for class 1 and none are also very similar.

Analysis of the confusion matrices for Feedforward-DBBE-BERT, Transformer-DBBE-BERT, and LSTM-DBBE-BERT in Figure 7 reveals that they are very similar. Most sentences with no or one elliptical node are correctly classified. Classification of sentences with 2 or more than 2 elliptical nodes remains a challenge, since most of them are misclassified. There is however an improvement in comparison to

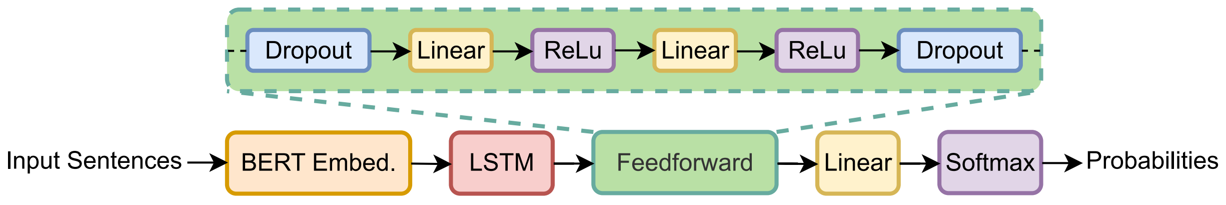

Figure 6: LSTM architecture.

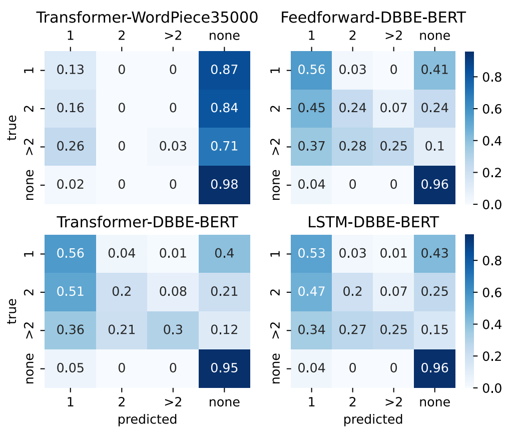

Figure 7: Confusion matrices for the models (without class weights) showing false negative rates and recall (test data set).

the baseline model: Feedforward-DBBE-BERT, for example, correctly classify 24% of 2-class sentences and 25% of >2-class sentences. The improvement of the models trained with the pretrained DBBE-BERT token embeddings is also confirmed by the precision, recall, and F1 scores per class reported in Table 3. I hypothesize that the bad performance in identifying sentences with two or more than two elliptical nodes much depends on the data set being highly unbalanced.

6 Conclusion

The present study compared five neural network models for prediction of the number of elliptical nodes in Ancient Greek sentences. The comparison showed that models with pretrained BERT token embeddings fine-tuned for the task at hand achieved the best results, with a good accuracy score (0.84), but a not very high macro-averaged F1 score (0.55 for Feedforward-DBBE-BERT). In comparison, the transformer architecture with randomly initialized token embeddings (Transformer-WordPiece35000) scored significantly worse, with a performance comparable to that of the baseline feedforward network with TF-IDF sentence vectorization (without class weights).

Model	Class	Precision	Recall	F1
BASELINE	1	0.38	0.12	0.18
	2	0.50	0.00	0.01
	>2	0	0	0
	none	0.80	0.97	0.88
FF-BERT	1	0.67	0.56	0.61
	2	0.47	0.24	0.32
	>2	0.52	0.25	0.34
	none	0.90	0.96	0.93
TR-BERT	1	0.64	0.56	0.59
	2	0.40	0.20	0.27
	>2	0.48	0.30	0.37
	none	0.90	0.95	0.93
LSTM-BERT	1	0.65	0.53	0.59
	2	0.43	0.20	0.28
	>2	0.46	0.25	0.32
	none	0.89	0.96	0.92

Table 3: Precision, recall, and F1 scores per class for the models Baseline, Feedforward-DBBE-BERT (FF-BERT), Transformer-DBBE-BERT (TR-BERT), and LSTM-DBBE-BERT (LSTM-BERT) (test data set).

The study also showed that performances of model architectures of different complexity fed with the same pretrained BERT token embeddings (i.e., Feedforward-DBBE-BERT, Transformer-DBBE-BERT, and LSTM-DBBE-BERT without class weights) proved to be comparable.

Limitations

Since annotation was performed by single annotators, some variability in ellipsis identification has to be expected in the original data.

Acknowledgements

This work has been supported by the German Research Foundation (DFG project number 408121292).

References

Johan Bos and Jennifer Spenader. 2011. An annotated corpus for the analysis of VP ellipsis. *Language Resources and Evaluation*, 45:463–494.

Giuseppe G. A. Celano. 2013. Argument-focus and predicate-focus structure in Ancient Greek: Word order and phonology. *Studies in Language*, 37(2):241–266.

Giuseppe G. A. Celano. 2023. *Opera Graeca Adnotata*. Version 0.1.0. Zenodo. https://doi.org/10.5281/zenodo.8158675.

Helma Dick. 1995. *Word Order in Ancient Greek: A Pragmatic Account of Word Order Variation in Herodotus*. J.C. Gieben, Amsterdam.

Livio Gaeta and Silvia Luraghi. 2001. Gapping in Classical Greek prose. *Studies in Language*, 25(1):89–113.

Vanessa B. Gorman. 2020. Dependency treebanks of Ancient Greek prose. *Journal of Open Humanities Data*, 6(1):1.

Daniel Hardt. 1997. An empirical approach to VP ellipsis. *Computational Linguistics*, 23(4):525–541.

Daniel Hardt. 2023. Ellipsis-dependent reasoning: a new challenge for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 39–47, Toronto, Canada. Association for Computational Linguistics.

Alek Keersmaekers, Wouter Mercelis, Colin Swaelens, and Toon Van Hal. 2019. Creating, enriching and valorizing treebanks of Ancient Greek. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 109–117, Paris, France. Association for Computational Linguistics.

Payal Khullar. 2020. Exploring statistical and neural models for noun ellipsis detection and resolution in English. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 139–145, Suzhou, China. Association for Computational Linguistics.

Raphael Kühner, Friedrich Blass, Bernhard Gerth, and William M. Calder. 1965. *Ausführliche Grammatik der Griechischen Sprache*. Wissenschaftliche Buchgesellschaft, Darmstadt.

Silvia Luraghi. 2003. Definite referential null objects in Ancient Greek. *Indogermanische Forschungen*, 108:167–194.

Leif Arda Nielsen. 2004. Robust VPE detection using automatically parsed text. In *Proceedings of the ACL Student Research Workshop*, pages 49–54, Barcelona, Spain. Association for Computational Linguistics.

Eduard Schwyzer. 1971. *Griechische Grammatik*, volume 2. C.H. Beck'sche Verlagsbuchhandlung, München.

Pranaydeep Singh, Gorik Rutten, and Els Lefever. 2021. A pilot study for BERT language modelling and morphological analysis for ancient and medieval Greek. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 128–137, Punta Cana, Dominican Republic (online). Association for Computational Linguistics.

Jeroen Van Craenenbroeck and Tanja Temmerman. 2019. *The Oxford Handbook of Ellipsis*. Oxford University Press, Oxford.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 1–11.

Wei-Nan Zhang, Yue Zhang, Yuanxing Liu, Donglin Di, and Ting Liu. 2019. A neural network approach to verb phrase ellipsis resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7468–7475.