



## Article

# Parsing Old English with Universal Dependencies—The Impacts of Model Architectures and Dataset Sizes

Javier Martín Arista , Ana Elvira Ojanguren López  and Sara Domínguez Barragán

Department of Modern Languages, Universidad de La Rioja, 26006 Logroño, LO, Spain;  
ana-elvira.ojanguren@unirioja.es (A.E.O.L.); sara.dominguez@aurea.unirioja.es (S.D.B.)

\* Correspondence: javier.martin@unirioja.es

## Abstract

This study presents the first systematic empirical comparison of neural architectures for Universal Dependencies (UD) parsing in Old English, thus addressing central questions in computational historical linguistics and low-resource language processing. We evaluate three approaches—a baseline spaCy pipeline, a pipeline with a pretrained tok2vec component, and a MobileBERT transformer-based model—across datasets ranging from 1000 to 20,000 words. Our results demonstrate that the pretrained tok2vec model consistently outperforms alternatives, because it achieves 83.24% UAS and 74.23% LAS with the largest dataset, whereas the transformer-based approach substantially underperforms despite higher computational costs. Performance analysis reveals that basic tagging tasks reach 85–90% accuracy, while dependency parsing achieves approximately 75% accuracy. We identify critical scaling thresholds, with substantial improvements occurring between 1000 and 5000 words and diminishing returns beyond 10,000 words, which provides insights into scaling laws for historical languages. Technical analysis reveals that the poor performance of the transformer stems from parameter-to-data ratio mismatches (1250:1) and the unique orthographic and morphological characteristics of Old English. These findings defy assumptions about transformer superiority in low-resource scenarios and establish evidence-based guidelines for researchers working with historical languages. The broader significance of this study extends to enabling an automated analysis of three million words of extant Old English texts and providing a framework for optimal architecture selection in data-constrained environments. Our results suggest that medium-complexity architectures with monolingual pretraining offer superior cost–benefit trade-offs compared to complex transformer models for historical language processing.

**Keywords:** syntactic parsing; old English; universal dependencies; natural language processing



Academic Editor: Alberto Abelló

Received: 28 May 2025

Revised: 17 July 2025

Accepted: 24 July 2025

Published: 30 July 2025

**Citation:** Martín Arista, J.; Ojanguren López, A.E.; Domínguez Barragán, S. Parsing Old English with Universal Dependencies—The Impacts of Model Architectures and Dataset Sizes. *Big Data Cogn. Comput.* **2025**, *9*, 199. <https://doi.org/10.3390/bdcc9080199>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. The UD Annotation of Old English

Universal Dependencies (UD) is an annotation framework developed for Natural Language Processing tasks, cross-linguistic comparison, translation, and language learning [1–3]. The UD framework provides a universal inventory of lexical categories, morphological features, and dependency relations suitable for cross-linguistic analysis that can also accommodate language-specific phenomena [4,5]. UD adopts a dependency-based syntactic representation, where binary asymmetric relations are established between heads and dependents [6]. The annotation framework is organised into three layers: universal part-of-speech tags (UPOS), morphological features (FEATS), and syntactic dependencies

(DEPREL). The UPOS layer consists of seventeen general lexical categories; the FEATS layer encodes morphological properties such as gender, number, case, and tense; and the DEPREL layer comprises a set of universal dependency relations that can be extended to handle language-specific constructions. Overall, UD prioritises universal linguistic patterns over language-specific ones, does not consider empty categories, and favours content words as syntactic heads over function words.

Old English (650–1150 CE) is a West Germanic language with a predominantly Germanic lexicon with borrowings from Latin and Old Norse. It is notable for its semantic transparency in word formation [7]; extensive inflection in nominal, pronominal, and verbal categories [8,9]; and relatively free word order compared to Modern English [10,11]. From the typological point of view, Old English is an SVO language in transition from the SOV type [12–17], which still surfaces in some dependent clauses and is reflected by other areas of grammar such as postposition or the genitive [18]. The written records of Old English amount to approximately three million words, preserved in around 3000 texts. The primary corpora for Old English are The Dictionary of Old English Web Corpus (3 million words; [19]) and The York–Toronto–Helsinki Parsed Corpus of Old English Prose (hereafter YCOE; 1.5 million words; [20]), the latter providing POS tagging and syntactic parsing for roughly half of the extant texts.

Recent research has applied the UD framework to the annotation of Old English. Martín Arista [21–23] establishes the foundations for parsing Old English within UD and extends the annotation framework to include word formation processes. This extension reflects the syntactic regularities and overlaps found in derivational processes and nominalisations in Old English, particularly those that inherit verbal properties [24]. As regards automatic dependency annotation, Villa and Giarda [25] evaluate the performance of a multilingual parser for Old English. Their study shows that combining Old English data with data from German and Icelandic yields the highest accuracy, with a peak performance of 75% accuracy for datasets combining Icelandic, German, and Old English. Villa and Giarda attribute these relatively low accuracy levels to linguistic factors such as Old English word order and case syncretism. These authors identify areas of error such as postpositions and discontinuity in relative clauses. They also note that inaccurate part-of-speech tagging leads to errors in dependency relations such as coordinating conjunctions, negation adverbial modifiers, auxiliaries, and locative and temporal adverbial modifiers. The work by Villa and Giarda [25] is discussed in more detail in Section 4, which compares their methods and results with this research.

Against this background, this paper focuses on the evaluation of UD parsing, with a specific focus on assessing how different neural architectures perform on the processing of Old English and on gauging the impact of the dataset sizes. The paper is structured as follows: Section 2 describes the different pipeline architectures and datasets of this study. Section 3 evaluates the performance of the models and the corpora with respect to the components of the pipeline. Section 4 compares our results with the state of the art in Natural Language Processing in general and with the automatic parsing of Old English in particular. Section 5 draws the main conclusions of the study.

The relevance of this study extends beyond the immediate application to Old English because it addresses questions of computational historical linguistics and low-resource language processing. Firstly, this research provides the first systematic empirical comparison of neural architectures specifically designed for Old English dependency parsing, considering that previous work has focused primarily on cross-lingual transfer learning approaches [25]. Secondly, this study contributes methodologically by demonstrating that monolingual pretraining on historical texts can outperform more complex architectural solutions. This challenges the assumption that transformer-based models universally provide

superior performance. This finding also has implications for the computational processing of historical and under-resourced languages, where data scarcity constrains the applicability of modern NLP techniques. Thirdly, from a practical point of view, the development of effective parsing tools for Old English carries out the automated analysis of approximately three million words, which amounts to the total written records and potentially accelerates research in historical syntax, corpus linguistics, and diachronic language change. The computational cost–benefit analysis provided here establishes evidence-based guidelines for researchers working with similar constraints, given that it offers a framework for selecting appropriate architectures based on available resources and performance requirements. Finally, this work contributes to our understanding of the scaling laws governing neural language models in low-resource scenarios, as it provides empirical evidence for the diminishing returns of architectural complexity when training data are limited.

## 2. Models and Data of the Study

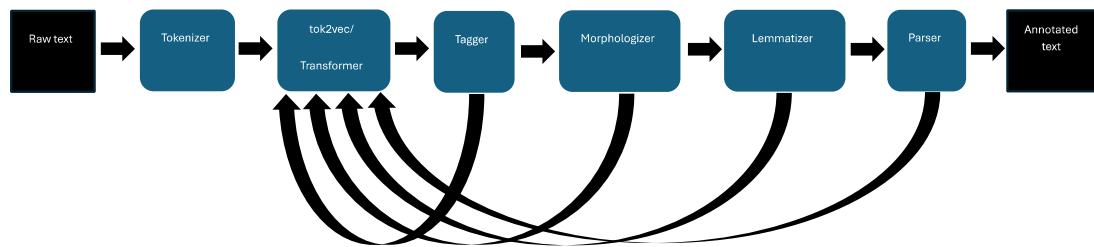
Three models have been trained on four dataset sizes, which are described in the remainder of this section. The first model is a basic pipeline with a default configuration that uses the spaCy default tok2vec component initialised with random weights. The second model also uses the tok2vec component but initialises its weights through a pretraining phase on an unannotated Old English corpus of about three million words. The third model is based on training from scratch with approximately 17 MB of text. Then, the tok2vec component is replaced with a custom-trained MobileBERT transformer.

The MobileBERT architecture (25.3 million parameters) was selected to match the limited size of available Old English training data. However, transfer learning from contemporary English BERT models was not viable for the reasons of lexical distance between Present-Day English and Old English, which has a consistently Germanic word stock; and spelling differences with respect to the contemporary language, which has lost certain graphemes (<æ/Æ>, <3/3>, <ð/Ð>, <þ/Þ>, and <ƿ/ƿ>) that English models cannot handle.

The pipeline architecture adopted in the test is based on the NLP library spaCy. It consists of six major components or stages, each handling specific aspects of the processing of Old English texts, which can be seen in Figure 1. The first stage is the Tokenizer, a rule-based component that splits text into tokens using predefined English rules. Unlike other components, the Tokenizer is non-trainable and serves as the initial stage that converts plain text into the internal data structure required by spaCy. Following tokenization, the second stage implements either a tok2vec or transformer component, both of which transform tokens into numerical vectors. Given the limited size of the dataset, this component is shared across subsequent stages to reduce the number of trainable parameters. Two implementations have been tested: the standard tok2vec and a MobileBERT transformer. The middle layers of the pipeline consist of the Tagger, which assigns POS tags (XPOS column), and the Morphologizer, responsible for UPOS and FEATS assignments. These components incorporate the vector representations provided by the previous stage. The pipeline continues with the trainable Lemmatizer component for LEMMA assignments, followed by the Parser, which handles both dependency parsing (HEAD and DEPREL assignments) and sentence boundary detection.

Thus described, the implementation maintains the pipeline architecture of spaCy, in which components can be trained independently or jointly. The pipeline also adopts the standard training workflow of spaCy, with configurable batch sizes (set to process at least 100 words per batch) and evaluation intervals (every 200 iterations). The evaluation metrics (TAG\_ACC, POS\_ACC, MORPH\_ACC, LEMMA\_ACC, DEP\_UAS, DEP\_LAS, and SENTS\_F) are all standard spaCy metrics and are calculated with built-in evaluation functions. Additionally, the loss tracking capabilities during training are used to assess the

performance of individual components. Separated loss values are obtained for the tok2vec, Tagger, Morphologizer, Lemmatizer, and Parser stages.



**Figure 1.** Pipeline stages.

The source of the datasets is ParCorOEv3. An open access annotated parallel corpus Old English–English [26]. The choice of curated Old English text includes *Ælfric’s Catholic Homilies I*, *The Anglo-Saxon Chronicle A*, *Anglo-Saxon Laws*, *St. Mark’s Gospel*, and King Alfred’s *Orosius*. The data of the test have been structured in four datasets of different sizes: 1000, 5000, 10,000, and 20,000 words for training. These increasing sizes have been established with a view to gauge the relationship between the training data volume and model performance. This is a fundamental aspect, considering the scarcity of Old English written records and annotated corpora noted above. An independent evaluation corpus (20%), which has been fully segregated from the training data, has provided benchmarking throughout the test. For pre-training purposes, we have used a larger unannotated corpus of approximately 3 million words (*The Dictionary of Old English Corpus*; [19]). The training and test datasets are described by words, tokens, and sentences in Table 1.

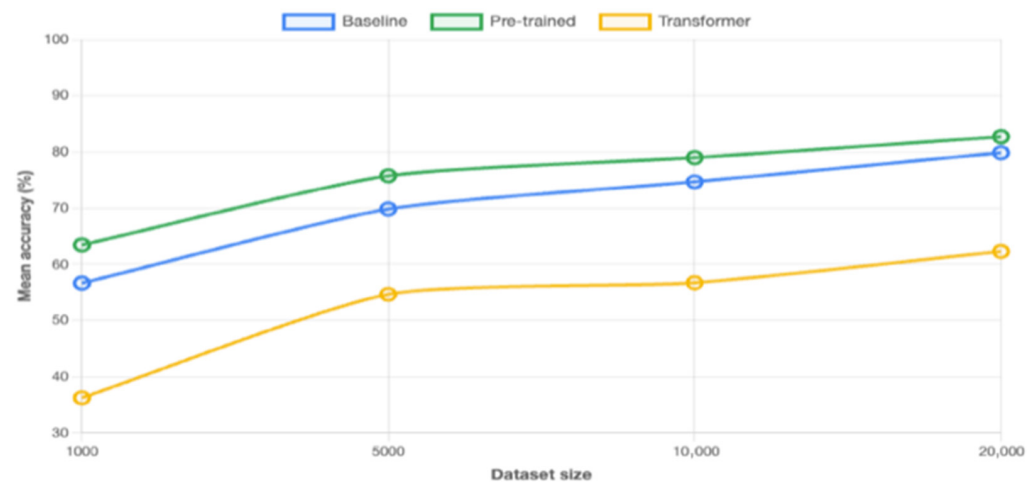
**Table 1.** Training and test datasets.

|              | Training | Test | Total  |
|--------------|----------|------|--------|
| 1000 words   |          |      |        |
| Tokens       | 995      | 4987 | 5982   |
| Sentences    | 59       | 288  | 347    |
| 5000 words   |          |      |        |
| Tokens       | 4992     | 4887 | 9879   |
| Sentences    | 283      | 288  | 571    |
| 10,000 words |          |      |        |
| Tokens       | 9982     | 4887 | 14,969 |
| Sentences    | 562      | 288  | 850    |
| 20,000 words |          |      |        |
| Tokens       | 19,991   | 4887 | 24,978 |
| Sentences    | 1134     | 288  | 1422   |

### 3. Performance Evaluation

Performance evaluation consists of multiple accuracy metrics for each component. The performance of the Tagger has been measured through TAG\_ACC for XPOS tagging. The performance of the Morphologizer has been tracked via POS\_ACC (UPOS) and MORPH\_ACC (FEATS). The performance of the Lemmatizer has been evaluated through LEMMA\_ACC. Dependency parsing has been gauged through both unlabelled (DEP\_UAS) and labelled (DEP\_LAS) attachment scores. Sentence boundary detection has been evaluated using the SENTS\_F metric. LAS (Labelled Attachment Score) and UAS (Unlabelled Attachment Score) are standard evaluation metrics in dependency parsing [27–29]. Whereas the UAS measures the percentage of tokens that are assigned the correct syntactic head, the LAS represents the percentage of tokens that are assigned both the correct syntactic head

and the correct dependency label. As a more stringent measure, the LAS is always equal to or lower than the UAS. Figure 2 presents the results of mean accuracy across all metrics by dataset size. The mean is calculated from the TAG, POS, FEATS, LEMMA, UAS, LAS, and SENT-F metrics (see Appendix A for a breakdown of scores by metric, architecture, and dataset).



**Figure 2.** Mean accuracy across all metrics by dataset size.

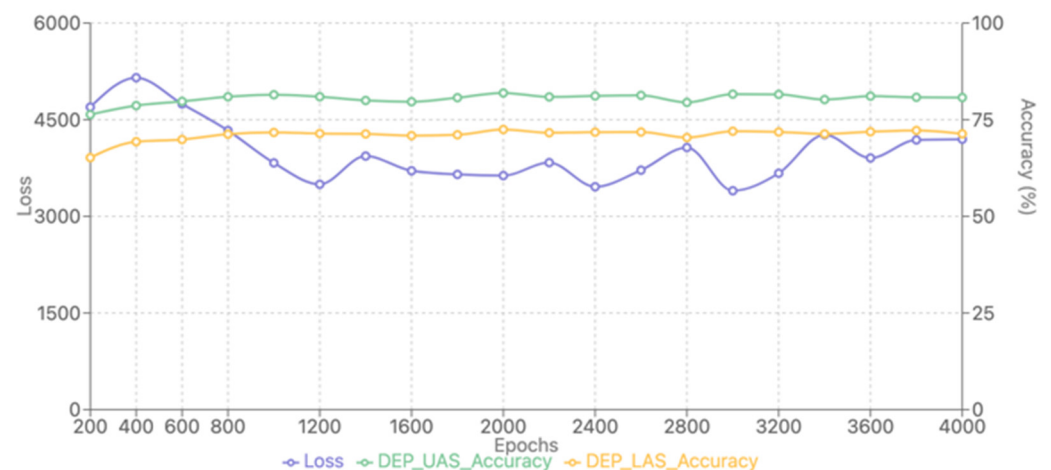
The performance analysis of this study reflects different degrees of task difficulty. Basic tagging tasks achieve 85–90% accuracy, while morphological analysis reaches around 80% accuracy. Dependency parsing obtains lower metrics, with accuracy around 75%. The pretrained model performs better across all metrics and corpus sizes. The transformer-based model obtains lower metrics.

The first model, using the default configuration of spaCy, establishes a baseline for performance. The model performs well on basic tokenization and simple POS tagging, with accuracy rates around 70–75% for basic POS tagging, but turns out significantly lower accuracy rates for more complex tasks like dependency parsing (see Appendix A). The pretrained model shows more promising results. By pretraining the tok2vec component on the larger unannotated corpus, we obtain notable improvements, with POS tagging accuracy increasing to 85–90% and dependency parsing displaying higher accuracy, around 75–80% (see Appendix A). The third model, implementing a transformer-based architecture using MobileBERT, constitutes an attempt to leverage advances in neural language models for historical language processing. This model seemed promising as it might capture long-range relations and dependencies blurred by the flexible word order of Old English, but it was hampered by the limited size of the training corpus (see Appendix A).

The accuracy metrics show interesting patterns across the three models. Sentence boundary detection turns out comparatively low metrics, probably due to the inconsistent punctuation patterns characteristic of Old English texts. Dependency parsing presents more variable results, although the pretrained model performs better across all corpus sizes. In POS tagging, accuracy improves with larger training sets, but the rate of improvement decreases significantly after the 10,000-word mark. Loss tracking during training reveals quick initial improvements followed by diminishing returns, which are particularly evident in the larger corpus sizes. Figure 2 illustrates the learning dynamics of the model. The results correspond to the best-performing architecture (pretrained model) and dataset (20,000 words).

As can be seen in Figure 3, there is clear convergence and consistent performance across key metrics, including loss, UAS, and LAS. To begin with, the loss sharply decreases initially. This is indicative of a rapid improvement in the optimisation of the model during the early

training phase. By approximately 2000 iterations, the loss stabilizes. This demonstrates that the model has converged to a steady state with minimal further improvement in optimisation. At the same time, both UAS and LAS increase rapidly during the initial iterations, which also reflects substantial gains in the accuracy for syntactic parsing. While UAS consistently outperforms LAS, both metrics stabilise at approximately 60% and 70%, respectively. The existence of this plateau is telling us that the ability of the model to assign correct dependencies and labels has reached its peak performance with the architecture and the dataset selected.



**Figure 3.** Loss, UAS, and LAS learning curves for the pretrained tok2vec model trained on the 20,000-word dataset, showing convergence patterns across dependency parsing metrics.

#### 4. Discussion

This section discusses our Old English parsing results from three angles: comparative performance, dataset scaling effects, and computational efficiency. First, we contextualise our findings against modern NLP benchmarks and compare them with Villa and Giarda’s [25] recent work on Old English dependency parsing. This comparison examines methodological differences in data selection, learning approaches, and performance outcomes. Second, we analyse how increasing the dataset size affects parsing accuracy across different model architectures, thus identifying optimal data thresholds and diminishing returns. Finally, we assess the computational requirements of each model architecture, providing a cost–benefit analysis that balances performance gains against resource consumption.

We address the question of compared performance metrics in the first place. The current state of the art in POS tagging reaches 97–98% accuracy across most languages [30], while up-to-date morphological analysers show an accuracy of 90–92% for morphologically rich languages [31]. Modern lemmatizers typically reach 95–97% accuracy [32] and sentence segmentation systems present F1-scores of 95% [33]. The dependency parsing of natural languages achieves UAS scores of 95–97% and LAS scores of 93–95% [34].

In the processing of a historical language like Old English, the results obtained so far do not reach the standards of natural languages for the reasons mentioned above. A relevant contribution to this field so far has been recently published by Villa and Giarda [25], who also explore the methodologies of parsing Old English within the UD framework, although their work differs significantly from the present study as to dataset construction, training methodologies, and syntactic analysis. Beginning with datasets, Villa and Giarda’s work focuses on a small, manually annotated dataset derived from two religious prose texts: *Adrian and Ritheus* and *Ælfric’s Supplemental Homilies*. Their corpus comprises 292 sentences (5315 tokens) converted from the YCOE into CoNLL-U format. To compensate for data scarcity, they employed cross-lingual transfer learning, training UUParser v2.4 on

combinations of Old English data and treebanks from three modern Germanic languages. The authors reduced support language treebanks to 60k tokens to avoid bias from larger datasets. In contrast, this study uses a larger and more diverse dataset (25k words) sourced from multiple Old English texts, including chronicles, as well as historical, religious, legal, and biblical texts. Unlike Villa and Giarda, this study trains models from scratch using spaCy pipelines and tests them.

Turning to learning methods and training approaches, Villa and Giarda's methodology involved multilingual transfer learning, which is based on the hypothesis that the structural similarities of related languages can improve parsing accuracy. Villa and Giarda trained their models on Old English alone and in combination with Icelandic, German, and Swedish, both individually and collectively. Their best-performing model combined Old English with Icelandic (UAS 68.44%, LAS 58.70%), likely due to Icelandic's conservative morpho-syntax (comprising case marking as well as V2 syntax) and distinctive graphemes (<æ/Æ> and <ð/Ð>). However, they observed diminishing returns when adding German or Swedish, which they attributed to syntactic divergences (like rigid SVO order in Swedish vs. relative flexibility in Old English). Notably, their monolingual Old English model underperformed (UAS 60.79%, LAS 47.23%).

This study gives priority to monolingual pretraining and the dataset size. The pre-trained model presented in this study achieves the highest scores (UAS 83.24%, LAS 74.23% with 20k words) and outperforms the MobileBERT transformer, which struggles as consequence of the lexical and graphemic differences between Old English and Present-Day English. Villa and Giarda's lower overall scores (maximal LAS 58.70%) likely stem from their smaller training data and the heterogeneity of their support languages. In contrast, the larger dataset selected for this study enables better generalisation, particularly for morphosyntactic features like case (80.1% accuracy) and tense (83.8%). The models used in this study excel at local dependencies, including, for example, possessive determiners (90.4% accuracy), but often fail to capture long-distance relations like clausal complements (25.5% accuracy). Villa and Giarda [25], as well as this study, show a consistent gap between UAS and LAS, which reflects the difficulty of labelling dependency relations compared to identifying head-dependent attachments. Villa and Giarda's best model (Icelandic + Old English) achieved a 9.74-point gap (UAS 68.44% vs. LAS 58.70%), while the pretrained model of this study returns a 9.01-point difference (UAS 83.24% vs. LAS 74.23%).

Both studies identify non-projective dependencies as a major issue of automatic parsing. Villa and Giarda highlight discontinuous constituents, particularly in relative clauses, where the antecedent and relative pronoun are separated by adjuncts. This study reports 0% accuracy on non-projective structures, such as relative clauses, clausal modifiers, and conjunctions. These shortcomings can be attributed to the variable word order of Old English, which gives rise to crossing dependencies that are not compatible with transition-based parsing algorithms [35].

We examine the question of the impact of dataset sizes now. Across all training methods, a clear correlation between dataset size and parsing accuracy emerges, though with important differences as far as learning patterns and efficiency are concerned. The relationship between dataset size and parsing accuracy follows a non-linear pattern with diminishing returns. The most remarkable improvements occur in the transition from 1000 to 5000 words, with the baseline model showing gains of +11.61% UAS and +25.85% LAS. The pretrained model, which starts from a higher baseline, achieves significant improvements of +7.99% UAS and +28.49% LAS. Subsequent increases from 5000 to 10,000 words and from 10,000 to 20,000 words yield smaller improvements. The rate of improvement does not plateau even at 20,000 words, which suggests that the limit of useful training data has not yet been reached.

The relationship between dataset size and parsing accuracy observed in this study gives insights into the scaling behaviour of neural models for historical languages, though several aspects require further empirical studies that can establish scaling principles. The non-linear improvement pattern, with the most substantial gains occurring in the transition from 1000 to 5000 words, suggests that there exists a critical threshold below which neural models cannot effectively learn the underlying linguistic patterns of Old English syntax. To quantify these scaling relationships more precisely, the methodology established in contemporary scaling law research should be followed [36]. Such an analysis would extrapolate to estimate the dataset sizes required to achieve performance levels comparable to modern language parsing (typically 95%+ UAS). A preliminary examination of our results suggests that reaching 90% UAS might require training datasets in the range of 100,000–200,000 words, although this estimate requires validation through additional data collection and experimentation.

The differential scaling behaviour across linguistic tasks—with POS tagging achieving near-optimal performance at 5000 words whilst dependency parsing continues improving substantially beyond 20,000 words—reflects the intrinsic complexity hierarchy of these tasks. This pattern aligns with theoretical expectations from learning theory, where tasks with larger output spaces and more complex structural dependencies require proportionally more training data. However, the specific scaling coefficients for each task type in historical language contexts remain to be established through systematic experimentation. The diminishing returns observed beyond 10,000 words, while still showing improvement, raise important questions about the optimal allocation of annotation effort in historical corpus development. The cost–benefit analysis suggests that the marginal improvement per additional annotated word decreases substantially after this threshold, though the absolute performance gains remain meaningful for practical applications. Future research should investigate whether alternative data augmentation strategies, such as synthetic sentence generation or cross-period transfer from Middle English corpora, might provide more efficient paths to improved performance than continued annotation of Old English texts.

The three models tested in this study demonstrate different scaling characteristics. The pretrained model outperforms the alternatives at all dataset sizes, as it achieves 83.24% UAS and 74.23% LAS with 20,000 words. This architecture also shows better data efficiency because it reaches higher accuracy levels with smaller datasets than the baseline model with comparable performance. In contrast, the transformer-based approach consistently underperforms, reaching only 60.17% UAS and 45.51% LAS even with the largest dataset.

The substantial underperformance of the transformer-based model, despite its theoretical advantages and success in modern NLP applications, calls for a detailed technical analysis. Several interrelated factors contribute to this counterintuitive result, which casts doubts on transformer superiority in sequence modelling tasks.

The primary factor is the mismatch between model capacity and available training data. The MobileBERT architecture, whilst relatively compact compared to full-scale transformers, contains approximately 25.3 million parameters. However, our largest training dataset comprises only 19,991 tokens across 1134 sentences, which creates a parameter-to-data ratio that is orders of magnitude higher than recommended for stable transformer training. Kaplan et al. [36] demonstrate that optimal transformer performance requires careful scaling between model size and dataset size, with their scaling laws suggesting that models with millions of parameters require billions of training tokens to achieve optimal performance. Our parameter-to-token ratio of approximately 1250:1 falls far outside the recommended range. This inevitably leads to severe overfitting despite regularisation techniques.

The lexical and orthographic characteristics of Old English present additional challenges for transformer architectures trained from scratch. Unlike modern English trans-

formers that benefit from extensive pretraining on contemporary texts, our MobileBERT model must learn both the language model objective and the downstream parsing tasks simultaneously. Old English employs several graphemes ( $\langle \text{æ}/\text{E} \rangle$ ,  $\langle \text{ð}/\text{Ð} \rangle$ ,  $\langle \text{þ}/\text{P} \rangle$ , and  $\langle \text{p}/\text{P} \rangle$ ) that are absent from modern English, which calls for the creation of a custom tokenizer and vocabulary from our limited corpus. This process eliminates any potential benefit from transfer learning that might partially compensate for data scarcity. Furthermore, the morphological richness of Old English, with its extensive nominal and verbal inflection systems, creates a large vocabulary relative to the corpus size, which increases the sparse data problem.

The training dynamics observed in our transformer model provide additional insights into its poor performance. The high standard deviations reported in Tables 2–4 (ranging from 14.09% to 18.33% across dataset sizes) indicate unstable training, throughout which the model struggles to converge consistently between the different metrics. This instability is characteristic of transformer training on insufficient data, where the model alternates between underfitting and overfitting different aspects of the training signal. Further analysis would be required to examine the loss landscapes and gradient dynamics during training to fully characterise this instability.

The attention mechanism of the architecture can capture long-range dependencies in sufficient data regimes, although it may be counterproductive when training data are extremely limited. The self-attention layers require substantial amounts of data to learn meaningful attention patterns, and with insufficient examples, they may learn false correlations that do not generalise. The relatively free word order of Old English, which should theoretically benefit from attention mechanisms, instead presents additional complexity that the model cannot adequately learn from our limited training examples.

That said, different annotation tasks exhibit distinct learning trajectories across dataset sizes. POS tagging and morphological feature recognition reach near-optimal performance relatively quickly, in such a way that the pretrained model achieves over 90% accuracy for POS tagging at just 5000 words. These tasks benefit from the relatively closed sets of possible tags and the strong correlation between word form and morphological features in Old English. Dependency parsing continues to improve substantially across all dataset increments.

This assessment raises two further questions. Firstly, is the advantage of the pretrained model consistent across all tasks or driven by exceptional performance on just a few metrics? And, secondly, does the transformer model struggle uniformly on all tasks, or is its weak overall performance primarily driven by specific metrics? This could explain which tasks benefit most from pretraining, where the simple baseline model may be adequate for obtaining similar results and at what dataset sizes architectural differences become more important.

**Table 2.** Performance gap between the pretrained and baseline models.

| Metric | Pretrained | Baseline | Difference<br>(Percentage Points) | Relative<br>Improvement |
|--------|------------|----------|-----------------------------------|-------------------------|
| XPOS   | 93.20%     | 90.66%   | +2.54                             | +2.8%                   |
| UPOS   | 92.96%     | 90.64%   | +2.32                             | +2.6%                   |
| FEATS  | 84.21%     | 81.00%   | +3.21                             | +4.0%                   |
| LEMMA  | 79.83%     | 79.91%   | −0.08                             | −0.1%                   |
| UAS    | 83.24%     | 78.26%   | +4.98                             | +6.4%                   |
| LAS    | 74.23%     | 68.10%   | +6.13                             | +9.0%                   |
| SENT-F | 71.38%     | 70.57%   | +0.81                             | +1.1%                   |
| Mean   | 82.72%     | 79.88%   | +2.84                             | +3.6%                   |

**Table 3.** Architecture comparison.

| Metric | Transformer | Baseline | Difference<br>(Percentage Points) | Performance<br>Ratio |
|--------|-------------|----------|-----------------------------------|----------------------|
| XPOS   | 79.91%      | 90.66%   | −10.75                            | 88.1%                |
| UPOS   | 79.89%      | 90.64%   | −10.75                            | 88.1%                |
| FEATS  | 64.95%      | 81.00%   | −16.05                            | 80.2%                |
| LEMMA  | 65.58%      | 79.91%   | −14.33                            | 82.1%                |
| UAS    | 60.17%      | 78.26%   | −18.09                            | 76.9%                |
| LAS    | 45.51%      | 68.10%   | −22.59                            | 66.8%                |
| SENT-F | 40.07%      | 70.57%   | −30.50                            | 56.8%                |
| Mean   | 62.30%      | 79.88%   | −17.58                            | 78.0%                |

**Table 4.** Mean accuracy standard deviations.

| Model       | Dataset | Mean Accuracy | Standard Deviation |
|-------------|---------|---------------|--------------------|
| Baseline    | 1000    | 56.67%        | 16.16%             |
| Baseline    | 5000    | 69.82%        | 12.40%             |
| Baseline    | 10,000  | 74.70%        | 10.80%             |
| Baseline    | 20,000  | 79.88%        | 8.69%              |
| Pretrained  | 1000    | 63.45%        | 19.35%             |
| Pretrained  | 5000    | 75.77%        | 12.28%             |
| Pretrained  | 10,000  | 79.00%        | 10.32%             |
| Pretrained  | 20,000  | 82.72%        | 8.06%              |
| Transformer | 1000    | 36.26%        | 15.55%             |
| Transformer | 5000    | 54.66%        | 18.33%             |
| Transformer | 10,000  | 56.74%        | 16.93%             |
| Transformer | 20,000  | 62.30%        | 14.09%             |

To answer the question on the consistency of the advantage of the pretrained model's performance, we analyse the performance gap between the pretrained model and the baseline model across all metrics at the 20K dataset size. The results are tabulated in Table 2.

As shown in Table 2, the advantage of the pretrained model is not uniform across all tasks. The largest advantages are in dependency parsing metrics (LAS +6.13 pp, UAS +4.98 pp). The pretrained model actually performs slightly worse on lemmatization (−0.08 pp) and its advantage is minimal for sentence segmentation (+0.81 pp). It seems to be the case that the pretrained model overall advantage is a consequence of its strong performance in syntactic tasks, particularly dependency parsing.

In order to answer the question on the uniformity of the struggle of the transformer model across tasks, we compare the transformer model to the baseline model at the 20k dataset size. The results are displayed in Table 3.

As can be seen in Table 3, the underperformance of the transformer model is non-uniform across tasks. Its worst result is for sentence segmentation (SENT-F), at only 56.8% of the baseline model's performance. Dependency parsing shows the deepest gap (LAS at 66.8% of the baseline model), while POS tagging shows the smallest difference (88.1% of the baseline model). The weak overall results of the transformer model, therefore, seem to be a direct consequence of its performance on syntactic parsing and sentence segmentation.

Table 4 shows the standard deviations across metrics for the mean accuracy.

As presented in Table 4, all models become more consistent as the dataset size increases because standard deviations decrease. This indicates that more data not only improve performance but also make performance more uniform across tasks. The transformer model has the highest standard deviations at larger dataset sizes. This highlights an uneven performance across tasks even with more data. This is in contradistinction to the pretrained model, which has the lowest standard deviations at the largest dataset size, which suggests that it achieves the most balanced performance across all metrics.

These remarks on the impact of dataset sizes add a new perspective to the main takeaway of this study, which underlines the adequacy for Old English parsing of the architecture based on the spaCy-based pipeline with pretraining on raw text. However, computational costs represent a crucial consideration alongside performance metrics when evaluating NLP architectures [32,37]. As a matter of fact, the three architectures under analysis have profiles that require different computational resources. They are illustrated with respect to the 20,000-word dataset in Table 5.

**Table 5.** Computational costs.

|                              | Base-Line | Pretrained                 | Transformer | Pretrained Advantage    | Transformer Gap              |
|------------------------------|-----------|----------------------------|-------------|-------------------------|------------------------------|
| <b>Metrics</b>               |           |                            |             |                         |                              |
| Mean                         | 79.8%     | 82.72%                     | 62.30%      | +2.84 pp<br>(+3.6%)     | −17.58 pp<br>(−22.0%)        |
| Standard deviation           | 8.69%     | 8.06%                      | 14.09%      | −0.63 pp<br>(−7.2%)     | +5.40 pp<br>(+62.1%)         |
| <b>Requirements</b>          |           |                            |             |                         |                              |
| Training time (relative)     | 1×        | 1–2×<br>(plus pretraining) | 5–10×       | 1–2× slower             | 5–10× slower                 |
| Inference speed (tokens/sec) | 1000+     | 800–1000                   | 100–300     | 10–20% slower           | 70–90% slower                |
| Memory usage (GB)            | 2–4       | 4–8                        | 8–16+       | 2–4× higher             | 4–8× higher                  |
| Model size (MB)              | 50–200    | 200–500                    | 500–1000+   | 2–5× larger             | 5–20× larger                 |
| GPU                          | Optional  | Recommended                | Required    | Higher hardware demands | Strict hardware requirements |
| Power                        | Low       | Medium                     | High        | 2–3× higher             | 5–10× higher                 |
| Cloud computing costs        | \$        | \$\$                       | \$\$\$      | 2× more expensive       | 6–10× more expensive         |

The comparative analysis of the three models on the 20,000-word dataset presents distinct trade-offs between performance, resources, and practicality. The pretrained model delivers the highest overall accuracy and requires moderately increased computational resources (1–2× training time and 2× cloud computing costs). This improvement is particularly pronounced in syntactic tasks. The baseline model offers excellent efficiency because it consumes minimal resources but achieves solid performance (79.88% average), making it the most accessible option for resource-constrained environments. This contrasts with the transformer model, which clearly underperforms both alternatives (62.30% average and 17.58 percentage points below baseline) despite demanding higher computational resources (5–10× longer training time and 70–90% slower inference).

For future applications, the pretrained approach represents the optimal balance between accuracy and efficiency for most tasks, particularly when both performance and computational viability matter. However, the strong performance-to-cost ratio of the baseline model makes it an attractive alternative, especially for basic NLP tasks with limited resources. This advantage becomes even more outstanding when working with larger datasets, where simpler models with sufficient data can approach the performance of more complex models. For under-resourced languages or strict computational constraints, the simpler pipeline might be preferable despite its slightly lower accuracy. The poor performance and high resource demands of the transformer makes it unsuitable for this NLP task.

Although this study focuses on supervised learning within traditional neural architectures, the issues encountered through the analysis advise us to consider alternative learning paradigms in future research. These paradigms, including reinforcement learning with human feedback, meta-learning, and advanced self-supervised learning, offer complementary solutions for the data scarcity problem in historical language processing. These three paradigms could be integrated within a unified framework that maximises the utility of both annotated and unannotated data.

Reinforcement learning (RL) approaches, particularly those incorporating human feedback mechanisms, could be used for addressing the non-projective dependency parsing shortcomings identified in both this study and the study by Villa and Giarda [25]. Recent advances in reinforcement learning from human feedback (RLHF) have demonstrated significant potential in structured prediction tasks. In this line, Wong et al. [38] show that crowd-sourced human feedback can guide RL training for code generation by large language models through Bayesian optimisation. Such approaches could be adapted for historical language parsing, where expert linguistic annotations could serve as reward signals for training dependency parsers on Old English texts. The application of policy gradient methods to dependency parsing decisions, guided by crowd-sourced feedback from historical linguists, might allow models to learn from weak supervision signals and to effectively deal with the annotation scarcity that constrains traditional supervised approaches. However, adapting RLHF methodologies to historical language processing would require the careful design of reward functions that capture linguistic validity and the development of efficient feedback collection mechanisms adapted to the specialised knowledge required for Old English syntactic analysis.

Complementing RLHF approaches, meta-learning and few-shot learning paradigms cope with data scarcity through cross-linguistic transfer mechanisms. Model-agnostic meta-learning (MAML) approaches [39] could use syntactic patterns learned from Germanic languages to improve Old English parsing with minimal fine-tuning of the data. Implementation would require developing language-specific adaptation modules that can distinguish between Germanic syntactic features (such as case marking patterns) and Old English-specific phenomena (such as non-nominative subjects). The meta-learning framework would require support and query sets, including sampling from different historical periods or dialectal variants, to guarantee generalisation.

Self-supervised learning beyond the tok2vec pretraining employed in this study represents a third complementary approach to the annotation of historical texts. Advanced masked language modelling could be specifically designed for Old English that incorporates structured masking strategies targeting inflectional patterns and compound word formation. Contrastive learning approaches could focus on diachronic variation and learn representations that capture both historical continuity and linguistic change. Syntactic pretraining tasks, such as constituency parsing or dependency relation assignment, could be adapted to Old English word order variation through multi-task learning frameworks that jointly meet morphological and syntactic objectives. However, implementing these approaches would require the development of Old English-specific tokenisation strategies that can handle unique graphemes, design of data augmentation techniques that preserve historical linguistic authenticity, and definition of metrics for variation in historical texts.

## 5. Conclusions

This study provides insights into the automatic parsing of Old English within the UD framework. Through a comprehensive evaluation of three pipeline architectures across four dataset sizes, we have identified effective approaches for processing historical language data.

The pretrained model consistently delivers superior performance. It achieves a 82.72% mean accuracy across all metrics with the largest dataset. This architecture excels particularly in syntactic tasks, with improvements of 6.13 and 4.98 percentage points in LAS and UAS, respectively, compared to the baseline model. The baseline model offers reasonable performance (79.88% mean accuracy) with minimal computational demands. This makes it suitable for resource-constrained environments. However, the transformer-based model significantly underperforms (62.30% mean accuracy), despite requiring more computational resources.

Our analysis of dataset impact shows that while larger training sets consistently improve performance, the relationship follows a non-linear pattern with diminishing returns. The most substantial gains occur when expanding from 1000 to 5000 words, whereas beyond this threshold, improvements are modest. Nevertheless, even at 20,000 words, the learning curve has not plateaued completely, which suggests that additional training data could yield further improvements.

The computational cost analysis points to relevant relations between model complexity, resource requirements, and performance. The pretrained model represents the optimal balance because it delivers the highest accuracy with moderate resource demands. For applications where computational efficiency is important, the baseline model offers an excellent alternative with acceptable performance.

Overall, our findings demonstrate that medium-complexity architectures with pre-training on raw text currently provide the best approach for the automated analysis of Old English dependencies, as they outperform both simpler baselines and more complex transformer-based models.

Several limitations of this study point towards areas requiring further research. The absence of confidence intervals and statistical significance testing for our performance metrics limits the strength of conclusions that can be drawn about architectural differences, particularly given the relatively small evaluation corpus. Future research may employ bootstrap sampling or cross-validation approaches to establish confidence bounds and determine whether observed performance differences are statistically significant.

The error analysis presented here remains at a high level, focusing on aggregate metrics rather than detailed linguistic phenomena. A comprehensive analysis of parsing errors by syntactic construction type, morphological complexity, and sentence length would provide insights into the specific challenges facing automated Old English processing. Such an analysis would require a manual examination of model outputs and categorisation of error types, which should result in improvements in both architectural design and training methodologies.

The computational cost analysis, which is informative for resource planning, lacks the granular detail necessary for precise reproducibility. Future work should involve specific hardware specifications, memory usage patterns, and energy consumption metrics to guarantee meaningful comparisons of different computational environments. Additionally, the relationships between model size, training time, and final performance require more systematic exploration to identify trade-offs for different use cases and resource constraints.

**Author Contributions:** Conceptualization, J.M.A.; methodology, J.M.A. and A.E.O.L.; formal analysis, A.E.O.L. and S.D.B.; investigation, J.M.A. and A.E.O.L.; data curation, S.D.B.; writing—original draft preparation, S.D.B.; writing—review and editing, J.M.A. and A.E.O.L.; visualization, S.D.B.; supervision, J.M.A. and A.E.O.L.; project administration, J.M.A.; funding acquisition, J.M.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by AEI /10.13039/501100011033, grant number PID2023149762NB-100MCIN.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used in this research will be available at the institutional repository of the University of La Rioja (<https://investigacion.unirioja.es/grupos/30/publicaciones> (accessed on 23 July 2025)).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

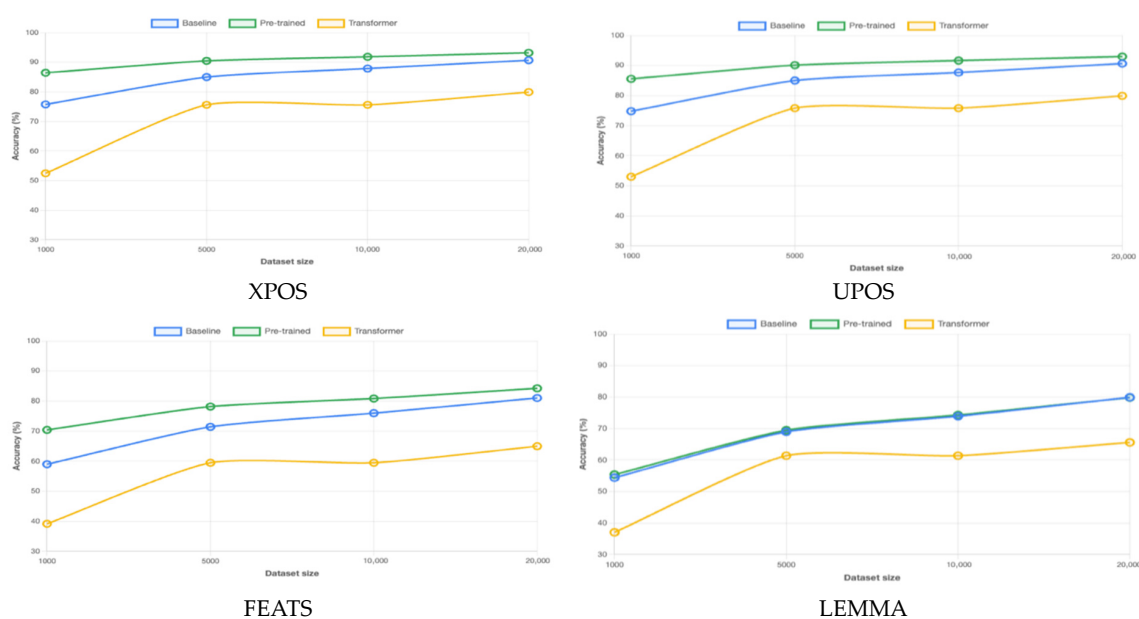
LAS    Labelled Attachment Score  
NLP    Natural Language Processing  
UAS    Unlabelled Attachment Score  
UD    Universal Dependencies

## Appendix A

The appendix presents a summary table and performance charts by component.

**Table A1.** Summary table.

| Model       | Dataset | Metrics (%) |       |       |       |       |       |        | Mean  |
|-------------|---------|-------------|-------|-------|-------|-------|-------|--------|-------|
|             |         | XPOS        | UPOS  | FEATS | LEMMA | UAS   | LAS   | SENT-F |       |
| Baseline    | 1000    | 75.75       | 74.79 | 58.96 | 54.39 | 57.44 | 27.32 | 48.07  | 56.67 |
| Baseline    | 5000    | 84.98       | 84.98 | 71.40 | 68.95 | 69.05 | 53.17 | 56.22  | 69.82 |
| Baseline    | 10,000  | 87.88       | 87.66 | 75.95 | 73.90 | 73.76 | 60.95 | 62.81  | 74.70 |
| Baseline    | 20,000  | 90.66       | 90.64 | 81.00 | 79.91 | 78.26 | 68.10 | 70.57  | 79.88 |
| Pretrained  | 1000    | 86.44       | 85.55 | 70.39 | 55.35 | 68.48 | 34.19 | 43.74  | 63.45 |
| Pretrained  | 5000    | 90.46       | 90.10 | 78.14 | 69.45 | 76.47 | 62.68 | 63.12  | 75.77 |
| Pretrained  | 10,000  | 91.84       | 91.62 | 80.82 | 74.28 | 80.07 | 69.10 | 65.27  | 79.00 |
| Pretrained  | 20,000  | 93.20       | 92.96 | 84.21 | 79.83 | 83.24 | 74.23 | 71.38  | 82.72 |
| Transformer | 1000    | 52.47       | 52.99 | 39.13 | 37.02 | 42.78 | 14.47 | 14.96  | 36.26 |
| Transformer | 5000    | 75.60       | 75.81 | 59.45 | 61.36 | 50.42 | 29.84 | 30.14  | 54.66 |
| Transformer | 10,000  | 75.60       | 75.81 | 59.45 | 61.36 | 55.58 | 38.83 | 30.56  | 56.74 |
| Transformer | 20,000  | 79.91       | 79.89 | 64.95 | 65.58 | 60.17 | 45.51 | 40.07  | 62.30 |



**Figure A1.** Cont.

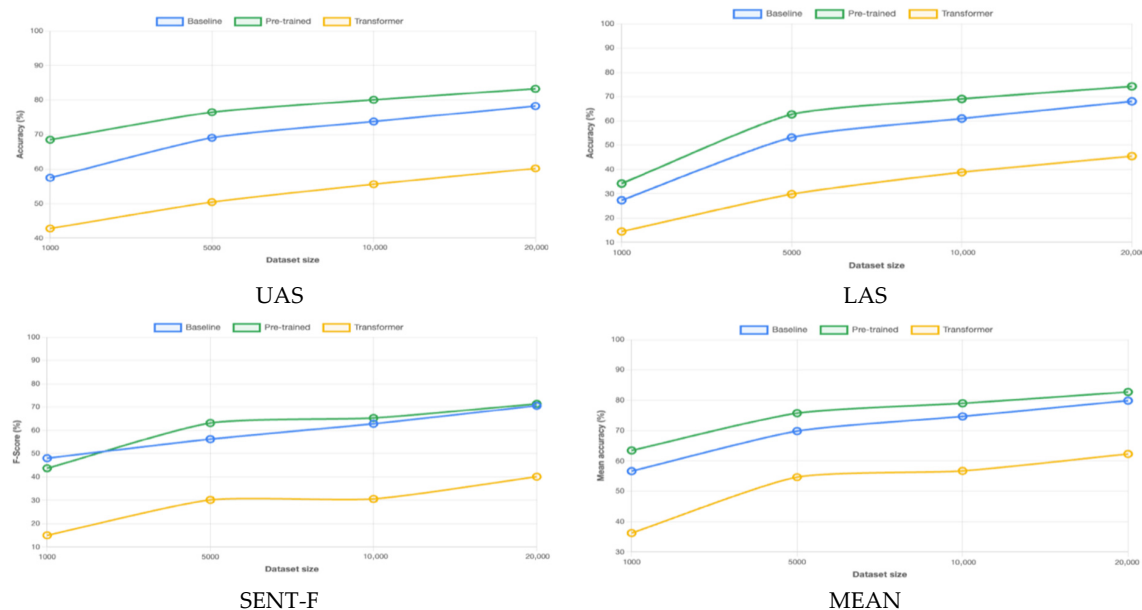


Figure A1. Performance by component.

## References

- Nivre, J.; de Marneffe, M.C.; Ginter, F.; Goldberg, Y.; Hajič, J.; Manning, C.D.; McDonald, R.; Petrov, S.; Pyysalo, S.; Silveira, N.; et al. Universal Dependencies v1: A multilingual treebank collection. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 1659–1666.
- Nivre, J.; de Marneffe, M.C.; Ginter, F.; Hajič, J.; Manning, C.D.; Pyysalo, S.; Schuster, S.; Tyers, F.; Zeman, D. Universal Dependencies v2: An evergrowing multilingual treebank collection. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 19 May 2020; pp. 4034–4043.
- Zeman, D. Universal Dependencies: Principles and practice. In *Computational Linguistics: Fundamentals and Advances in Natural Language Processing*; Elena, C., Lluís, M., Pierre, Z., Aurélie, N., Henning, W., Eds.; Springer: Berlin/Heidelberg, Germany, 2024; pp. 205–231.
- De Marneffe, M.C.; Dozat, T.; Silveira, N.; Haverinen, K.; Ginter, F.; Nivre, J.; Manning, C. Universal Stanford Dependencies: A cross-linguistic typology. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 26–31 May 2014; pp. 4585–4592.
- De Marneffe, M.C.; Manning, C.; Nivre, J.; Zeman, D. Universal Dependencies. *Comput. Linguist.* **2021**, *47*, 255–308. [\[CrossRef\]](#)
- De Marneffe, M.C.; Manning, C. *Stanford Typed Dependencies Manual*; Technical Report; Stanford University: Stanford, CA, USA, 2016.
- Kastovsky, D. Semantics and Vocabulary. In *The Cambridge History of the English Language I*; Richard, M.H., Ed.; Cambridge University Press: Cambridge, UK, 1992; pp. 290–408.
- Campbell, A. *Old English Grammar*; Oxford University Press: Oxford, UK, 1987.
- Middeke, K. *The Old English Case System: Case and Argument Structure Constructions*; Brill: Leiden, The Netherlands, 2022.
- Fischer, O.; van Kemenade, A.; Koopman, W.; van der Wurff, W. *The Syntax of Early English*; Cambridge University Press: Cambridge, UK, 2000.
- Ringe, D.; Taylor, A. *The Development of Old English*; Oxford University Press: Oxford, UK, 2014.
- Pintzuk, S. Phrase Structures in Competition: Variation and Change in Old English Word Order. Ph.D. Thesis, University of Pennsylvania, Philadelphia, PA, USA, 1991.
- Pintzuk, S. *Phrase Structures in Competition: Variation and Change in Old English Word Order*; Routledge: Oxfordshire, UK, 1999.
- Kroch, A.; Taylor, A. Verb-object order in Early Middle English. In *Diachronic Syntax: Models and Mechanisms*; Pintzuk, S., Tsoulas, G., Warner, A., Eds.; Oxford University Press: Oxford, UK, 2000; pp. 132–160.
- Koopman, W. Transitional syntax: Postverbal pronouns and particles in Old English. *Engl. Lang. Linguist.* **2005**, *9*, 47–62. [\[CrossRef\]](#)
- Haeberli, E.; Pintzuk, S. Revisiting verb (projection) raising in Old English. *York Pap. Linguist. Ser. 2* **2006**, *6*, 77–94.
- Van Kemenade, A. The syntax of Old English. In *A History of the English Language*; Hogg, R., Denison, D., Eds.; Cambridge University Press: Cambridge, UK, 2006; pp. 69–113.
- Allen, C. *Genitives in Early English: Typology and Evidence*; Oxford University Press: Oxford, UK, 2008.

19. Healey, A.; diPaolo Wilkin, J.P.; Xiang, X. *The Dictionary of Old English Web Corpus*; Dictionary of Old English Project, Centre for Medieval Studies, University of Toronto: Toronto, ON, Canada, 2004.
20. Taylor, A.; Warner, A.; Pintzuk, S.; Beths, F. *The York-Toronto-Helsinki Parsed Corpus of Old English Prose*; Department of Language and Linguistic Science, University of York: Heslington, UK, 2003.
21. Martín Arista, J. Old English Universal Dependencies: Categories, Functions and Specific Fields. In Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022), Online Streaming, Vienna, Austria, 3–5 February 2022; Volume 3, pp. 945–951.
22. Martín Arista, J. Toward the morpho-syntactic annotation of an Old English corpus with Universal Dependencies. *Rev. Lingüística Leng. Apl.* **2022**, *17*, 85–97. [\[CrossRef\]](#)
23. Martín Arista, J. Toward a Universal Dependencies Treebank of Old English: Representing the Morphological Relatedness of Un-Derivatives. *Languages* **2024**, *9*, 76. [\[CrossRef\]](#)
24. Ojanguren López, A.E. Structuring the Lexicon of Old English with Syntactic Principles: The Role of Deverbal Nominalisations with Aspectual and Control Verbs. In *Structuring Lexical Data and Digitising Dictionaries. Grammatical Theory, Language Processing and Databases in Historical Linguistics*; Martín Arista, J., Ojanguren López, A.E., Eds.; Brill: Leiden, The Netherlands, 2024; pp. 327–355.
25. Villa, P.; Giarda, M. Old meets new: Universal Dependencies for historical languages. *J. Lang. Technol. Comput. Linguist.* **2023**, *36*, 17–43.
26. Domínguez Barragán, S.; Fidalgo Allo, L.; García Fernández, L.; Hamdoun Bghiyel, Y.; Lacalle Palacios, M.; Mateo Mendaza, R.; Novo Urraca, C.; Ojanguren López, A.E.; Ruíz Narbona, E.; Torre Alonso, R.; et al. *ParCorOE3. An Open Access Annotated Parallel Corpus Old English-English*; Martín Arista, J., Ed.; Nerthus Project, Universidad de La Rioja: Logroño, Spain, 2023.
27. Nivre, J.; Hall, J.; Nilsson, J.; Chanev, A.; Eryigit, G.; Kübler, S.; Marinov, S.; Marsi, E. MaltParser: A language-independent system for data-driven dependency parsing. *Nat. Lang. Eng.* **2007**, *13*, 95–135. [\[CrossRef\]](#)
28. Kübler, S.; McDonald, R.; Nivre, J. *Dependency Parsing. Synthesis Lectures on Human Language Technologies*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2009.
29. Manning, C. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing*; Alexander, F.G., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 171–189.
30. Wang, Y.; Che, W.; Tian, J.; Liu, T. Improving Bidirectional Decoding with Dynamic Target Semantics in Neural Machine Translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, Bangkok, Thailand, 1–6 August 2021; pp. 5628–5638.
31. Şahin, G.G. Framing Neural Morphological Tagging for Low-Resource Languages. In Proceedings of the 28th International Conference on Computational Linguistics, Online, Barcelona, Spain, 8–13 December 2020; pp. 3513–3519.
32. Kanerva, J.; Ginter, F.; Salakoski, T. Universal Lemmatizer: A sequence-to-sequence model for lemmatizing Universal Dependencies treebanks. *Nat. Lang. Eng.* **2020**, *26*, 205–233. [\[CrossRef\]](#)
33. Augustyniak, Ł.; Morzy, M.; Kajdanowicz, T.; Kazienko, P.; Dąbrowski, M.; Żelasko, P. Punctuation prediction model for conversational speech. In Proceedings of the Interspeech 2020, Virtual Event, Shanghai, China, 29 October 2020; ISCA: Singapore; pp. 4911–4915.
34. Ahmad, W.U.; Peng, H.; Chang, K.-W. COLT5: Faster long-range transformers with conditional computation. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 11588–11608.
35. Jurafsky, D.; Martin, J.H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, 2nd ed.; Prentice-Hall: Englewood Cliffs, NJ, USA, 2020.
36. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling laws for neural language models. *arXiv* **2020**, arXiv:2001.08361.
37. Rae, J.W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, F.; Aslanides, J.; Henderson, S.; Ring, R.; Young, S.; et al. Scaling language models: Methods, analysis & insights from training Gopher. *arXiv* **2021**, arXiv:2112.11446.
38. Wong, M.F.; Tan, C.W. Aligning crowd-sourced human feedback for reinforcement learning on code generation by large language models. *IEEE Trans. Big Data* **2024**, *1*–12. [\[CrossRef\]](#)
39. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.