Natural Language Engineering (2023), 1–30
doi:10.1017/S1351324923000384

ARTICLE

Creating a large-scale diachronic corpus resource: Automated parsing in the Greek papyri (and beyond)

Alek Keersmaekers and Toon Van Hal

Department of Linguistics, KU Leuven, Leuven, Belgium
Corresponding author: Alek Keersmaekers; Email: alek.keersmaekers@kuleuven.be

(Received 9 December 2020; revised 7 July 2023; accepted 7 July 2023)

Abstract

This paper explores how to syntactically parse Ancient Greek texts automatically and maps ways of fruitfully employing the results of such an automated analysis. Special attention is given to documentary papyrus texts, a large diachronic corpus of non-literary Greek, which presents a unique set of challenges to tackle. By making use of the Stanford Graph-Based Neural Dependency Parser, we show that through careful curation of the parsing data and several manipulation strategies, it is possible to achieve an Labeled Attachment Score of about 0.85 for this corpus. We also explain how the data can be converted back to its original (Ancient Greek Dependency Treebanks) format. We describe the results of several tests we have carried out to improve parsing results, with special attention paid to the impact of the annotation format on parser achievements. In addition, we offer a detailed qualitative analysis of the remaining errors, including possible ways to solve them. Moreover, the paper gives an overview of the valorisation possibilities of an automatically annotated corpus of Ancient Greek texts in the fields of linguistics, language education and humanities studies in general. The concluding section critically analyses the remaining difficulties and outlines avenues to further improve the parsing quality and the ensuing practical applications.

Keywords: Parsing; Ancient Greek; Corpus applications

1. Introduction

Today, growing attention is paid to Natural Language Processing (NLP) applications tailored to ancient languages, both by the NLP community, which has only recently begun to engage with historical data, and by classicists and ancient historians, who are starting to embrace their increasing possibilities of conducting both large-scale and fine-grained corpus-based work. As further explained in Section 6, morphologically, syntactically and possibly semantically annotated sentences are not only an asset to linguists. Treebanks will also prove useful for didactic and historical purposes, especially if high-quality automatic trees can be generated. It is against this backdrop that this contribution can be situated. It is therefore the aim of this paper (1) to investigate the extent to which we can make syntactic dependency trees of Ancient Greek texts automatically, with a special focus on the impact of the annotation format on parser achievements and (2) to map ways of fruitfully employing the results of such an automated analysis. The focus is on the subcorpus of documentary papyrus texts. Consisting of everyday texts such as letters, petitions and court proceedings from the third century BC to the eighth century AD, this corpus offers unique insights into the vernacular language at the time, while being far more diverse in sociolinguistic terms than

This contribution is in part a follow-up study of an unpublished master's thesis, supervised by the authors (Mercelis, 2019). We are very much indebted to the constructive and detailed remarks given by several anonymous reviewers.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

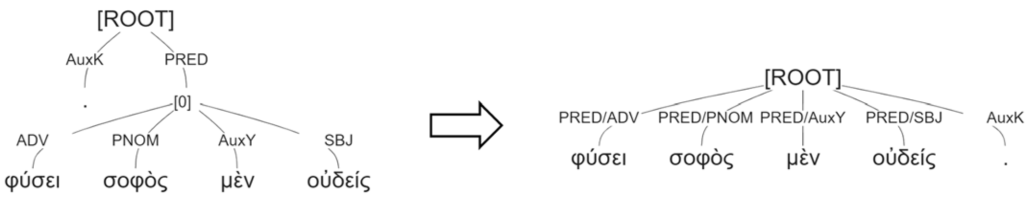


Figure 1. Base representational format for ellipsis (φύσει "by nature" σοφὸς "wise" μὲν (particle) οὐδείς "nobody" – "Nobody [is] wise by nature").

literary texts, which are typically written by male authors from the highest ranks of society (Fögen 2010: 312). Additionally, we will also discuss the possibilities of this approach for literary texts. This paper aims at a readership of both NLP specialists interested in the specifics of Ancient Greek and classicists with an interest in corpus-based approaches. What is self-evident to one audience is not automatically so to the other, and therefore, we opted to be as explicit as possible.

After a brief overview of the current state of the art regarding the automated syntactic processing for Ancient Greek in general and the papyri in particular (Section 2), this paper discusses the principal problems and challenges (Section 3). We then present the methodology and approach adopted in this study (Section 4), which precedes an in-depth discussion of the main results (Section 5). Subsequently, we briefly survey the possible applications of an automatically annotated corpus of Ancient Greek texts (Section 6). We conclude this contribution by critically analysing the remaining difficulties as well as outlining avenues for solving them (Section 7).

2. State of the field and related work

A treebank is a corpus whose sentences are syntactically annotated. This annotation is often visualised as a tree (cf. Figures 1–4 for some examples). Sentences can be syntactically analysed in divergent ways. The dependency model on the one hand and phrase-structure or constituency model on the other are often regarded as the two principal players, even though they should not be seen in a binary opposition, given that many hybrid representations are in use too (cf. Kübler, McDonald, and Nivre 2009: 3). In the phrase-structure model, which is predominant in generative linguistics, individual words are part of larger phrases (such as the noun phrase, verb phrase, etc.). This model assumes that the sentence's constituents can be neatly distinguished from each other. It is safe to say that phrase-structure grammars – and, as a result, annotation styles based on phrase-structure grammars – prevailed in linguistics until the end of the 20th century. This went hand-in-hand with the predominance of a rule-based approach in NLP. However, experiments in automatically analysing syntactic sentence structures soon revealed that such a model proved rather inadequate for languages with flexible word and constituent order (Kübler *et al.* 2009: 1). Dependency models turned out to cope better in this regard. In such models, no abstract phrases are distinguished. Instead, words are linked directly to each other, and each word is made dependent on only one other word. In this contribution, our focus is on Ancient Greek, which is a clear representative of a language with a free constituent order (Dik 1995). It therefore comes as no surprise that most recent treebank projects dealing with Ancient Greek are dependency based, even though there are also a number of phrase structure treebanks of Ancient Greek (see Robie 2017 for an overview). It is also important to emphasise that the linguistic characteristics of Ancient Greek that we will address in the rest of the paper are typical, but far from exclusive, to Ancient Greek.

The following overview of related work deals with annotation projects for Ancient Greek in general and is not limited to annotation work on the Ancient Greek papyri. Most efforts to syntactically process the Ancient Greek corpus have so far focused on **manual annotation**. The

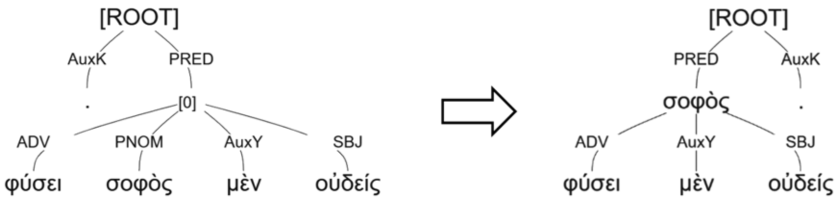


Figure 2. New representational format for elliptic copula (sentence: see Figure 1).

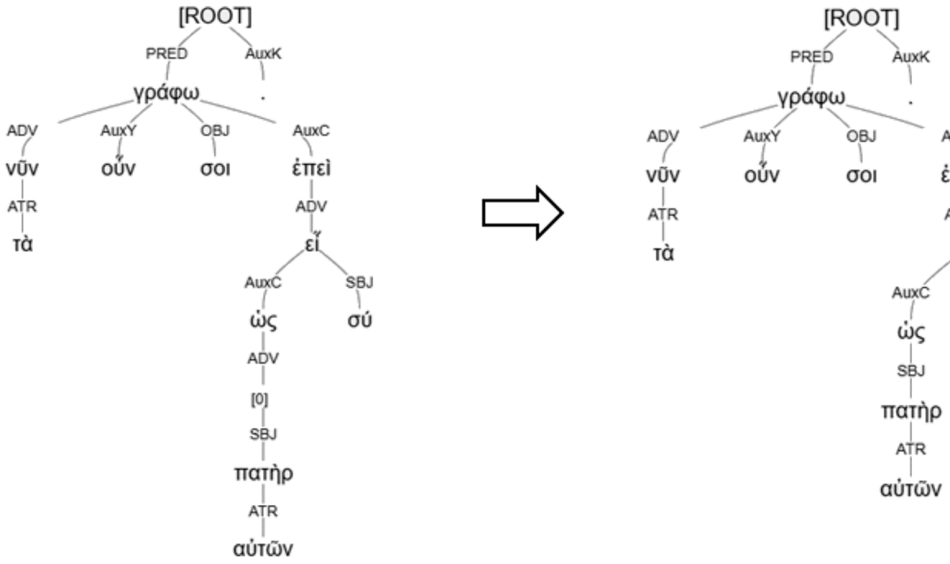


Figure 3. New representational format for comparative constructions (sentence: τὰ νῦν “now” οὖν “so” γράφω “I write” σοι “to you” ἐπεὶ “since” ὡς “like” πατὴρ “father” αὐτῶν “of them” εἶ “are” σύ “you” – ’So I write to you now since you are like their father“).

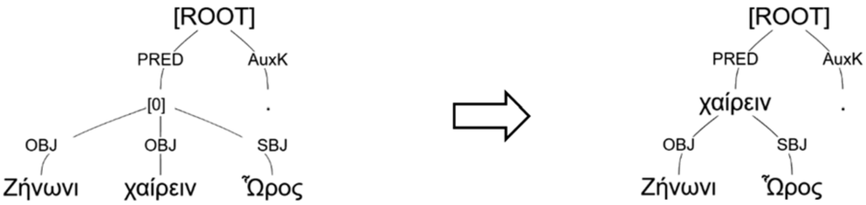


Figure 4. New representational format for infinitives without a main verb (sentence: Ζήνωνι “to Zenon” χαίρειν “be happy” Ὧρος “Horos” – “Horos [tells] Zenon to be happy,” i.e. “Horos greets Zenon“).

ongoing dependency treebank initiatives for Ancient Greek will be briefly described here (see also Keersmaekers *et al.* 2019). The two most extensive projects that exist to date are the Perseus Ancient Greek (and Latin) Dependency Treebanks (AGDT) (Bamman, Mambrini, and Crane 2009) and the PROIEL Treebank (Haug and Jøhndal 2008). The PROIEL treebanks, following an idiosyncratic annotation scheme, originally included an analysis of the entire New Testament (not only of Greek but also of several other languages). In addition, substantial parts of Herodotus were published in the same scheme, as well as a late Byzantine text by George Sphrantzes. Its set of syntactic labels is slightly more extensive, including, for instance, special labels for agents and indirect objects.

The AGDT consist of a substantial poetry part (both archaic poetry – Homer and Hesiod most notably – as well as classical poetry, with a special focus on Aeschylus and Sophocles), while the other treebanks are literary prose texts. These two parts are now gradually being placed under separate collections, managed by the two main annotators. A revamped version of the poetic texts will become available through the *Daphne* platform (Mambrini 2020). The majority of the AGDT prose texts were annotated by Vanessa Gorman, who published a sizeable number of treebanks in her own repository[a] too (with a special focus on history and oratory) Gorman (2020). In terms of treebank design, the AGDT is modelled on the Perseus Latin Dependency Treebank (Bamman *et al.* 2009: 6), which is in turn heavily based on the Prague Dependency Treebank (PDT) (Bamman and Crane 2006: 68–71). In its earliest versions, the PDT, which is concerned with the detailed linguistic annotation of Czech texts, consisted of a morphological layer and a so-called analytical layer (annotating dependency relations). Information added in later versions, such as a tectogrammatical layer (which annotates deep syntax and valency), information on coreference, information structure and multiword expressions (Hajič *et al.* 2018), are absent from the AGDT. For rendering possessive genitives, ablative absolutes and ellipsis, the Latin Dependency Treebank developed its own encoding strategies.

Both the AGDT and PROIEL collections have been converted into the Universal Dependencies scheme, which explain their present-day prominence. Through the framework of Universal Dependencies the international community is making progress in producing annotations of both morphology and syntactic relations (semantics is not included) in a consistent format with cross-linguistic validity. Currently, dependency treebanks for more than 100 languages, also including ancient languages such as Latin, Sanskrit and Classical Chinese, are freely distributed through the Clarin infrastructure (see universaldependencies.org). In this format, they serve as the Ancient Greek basis for testing new taggers and parsers in the CONLL shared tasks, a worldwide competition of NLP programmers (Zeman *et al.* 2018).

Next to the AGDT and PROIEL treebanks, the Helsinki-based *Sematia* (original name, now known as PapyGreek) initiative offers documentary papyri, following the AGDT annotation scheme with some minor modifications (Vierros and Henriksson 2017). It is also worthwhile to mention the Harrington Trees, containing – among other texts – Lucian's *True Histories* (Harrington 2018). Finally, for this project, we developed our own collection of syntactic trees, the Pedalion Treebanks (Keersmaekers *et al.* 2019), including classical and post-classical prose and poetry, with a special focus on genres and authors that are less well represented in the major treebanking projects – see Sections 4.1 and 6 for more details. This outline shows that, until now, there has been a strong emphasis on setting up treebank projects and establishing annotation conventions. In addition, valuable visualisation initiatives were also taken. In the early days, treebank annotators were bound to make their annotations directly in XML files, a way of proceeding that is error prone and anything but intuitive. Thanks to the Perseids and Arethusa initiatives, users are nowadays able to create a syntactic annotation of a sentence through dynamically visualised trees (Almas and Beaulieu 2016).

In contrast, there have only been limited efforts on **automated syntactic analysis** for Ancient Greek. The first and only study we are aware of that is specifically focused on Ancient Greek is Mambrini and Passarotti (2012), who trained and tested *MaltParser* (Nivre *et al.* 2007) on poetic data from the AGDT. When trained and tested on Homeric data (about 120K tokens for both train and test set), they attain a maximal Labeled Attachment Score (LAS) score of 0.717.[b] Other than that, Ancient Greek is sometimes used as a test case together with other languages. An example

[a] https://github.com/vgorman1/Greek-Dependency-Trees

[b] In dependency-based parsing analysis, the LAS is the most used evaluation metric, an accuracy-based measure representing the percentage of tokens that have both their syntactic head ('attachment') and relation ('label') correctly identified. Other frequent metrics are the UAS (Unlabeled Attachment Score, only the heads are evaluated) and the Label Accuracy Score (LS, ignoring the heads, while evaluating the labels).

of this is Lee *et al.* (2011) who, performing joint tagging/parsing, report increases in both tagging and parsing accuracy over a simple pipeline model (i.e. when part-of-speech tagging and syntactic parsing are done independently from each other). As the PROIEL and AGDT corpora are present in the Universal Dependencies (UD) project (Nivre *et al.* 2016), they are also sometimes evaluated together with other UD treebanks, such as in the 2018 and 2019 CONLL shared tasks on multilingual syntactic parsing, where the highest achieved LAS (with the HIT-SCIR parsing system) is 0.794 for the AGDT treebanks (about 160K training and 21K test tokens) and 0.793 for the PROIEL treebanks (about 187K training and 13K test tokens) (Zeman *et al.* 2018).[c] Such initiatives are strongly focused on the development of generic parsers that can be applied to a large group of languages. Having a different approach, this paper seeks to achieve better results for a single language, in this case Ancient Greek, by optimising the language data for automatic syntactic analysis.

While most research on the diachronic syntax of Greek has traditionally focused on the language of literary texts, the so-called Greek documentary papyrus corpus has recently received increasing attention. As a relatively large text corpus (about 4.5 million words), these texts are well-suited for a quantitative, corpus-based approach, as already suggested by Porter and O'Donnell (2010). However, up until recently there have been no concentrated efforts to process this corpus in order to enable linguistic queries in these texts. The papyri have been morphologically processed by Celano (2018) and Keersmaekers (2020b). The present paper relies on Keersmaekers (2020b) as a starting point, who achieved a ca. 95% accuracy in morphology in a pipeline model for the papyri (which includes tokenisation and lemmatisation as well, the latter with an accuracy rate of about 99%), in order to achieve high quality parsing results for Ancient Greek.[d] It is indeed vital to have a decent morphologically annotated base to start from, as there is a strong interaction between morphology and syntax (e.g. case usage, agreement, etc.: see Section 3.2). One could also perform morphological and syntaxis analysis jointly, as Lee *et al.* (2011) did – however, as Keersmaekers (2020b) has shown, at the moment high performing joint tagging and parsing systems that are suitable for Ancient Greek are still lacking. No efforts have been undertaken to parse the papyri syntactically so far, and since these texts substantially differ from literary texts (see also Section 3.3), our results will not be directly comparable with the previous results found for Ancient Greek that were discussed above.

3. Problems

As a large historical corpus of a highly inflectional language, the Greek papyrus corpus poses several problems for automated syntactic processing. This section gives an overview of the main problems we encountered, while the solutions we implemented are discussed in the next section.

3.1 Annotation problems

The above-mentioned treebanks result from the voluntary work of many individuals, whose uncoordinated work has inevitably led to a number of incoherent annotations. However, these inconsistencies are by no means solely due to errors. For instance, in most dependency models, each token depends on exclusively one head (Celano 2019: 280), while there are many cases in Greek where a constituent can be regarded as dependent on two (or more) different verbs

[c] This is on the lower end of parsing accuracy: the Ancient Greek AGDT and PROIEL treebanks rank 54th and 55th, respectively, in LAS of all 82 treebanks for the best performing parsing system (the treebank with the best LAS is the Polish LFG treebank, with an LAS of 0.949). This is especially remarkable given that the PROIEL and AGDT treebanks have the 18th and 24th biggest training set, respectively, of all treebanks in the contest.

[d] These experiments were carried out on a smaller dataset than the one in the current paper, of 2400 test tokens, and the NLP tools were trained on mainly literary data from various sources, 1 M tokens in total.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

simultaneously. For example, in sentence (1) below, we see that "daughter" is the object of both ἀπατήσας "deceive, seduce" and ἐξαγαγὼν "lead away" as well as the main predicate κρύπτει "hide" simultaneously. In Ancient Greek, it frequently happens that a predicate and a dependent participial clause share the same object (see Luraghi 2003 for more details). However, in the annotation only one head can be selected. Hence, very often, several solutions and approaches can be defended. This does not alter the fact that, from several perspectives, consistency is key: not only for improving the quality of natural language processing systems but also for evaluating the test data and for conducting corpus-based research in the treebanks.

(1) Δημήτρ[ι]ος δέ μου ὁ ἀμπελουργὸς ἀπατήσας τὴν
Demetrios.nom prt I.gen the.nom vinedresser.nom **deceive.ptc.nom.sg** the.acc
θυγατέρα ἐξαγαγὼν κρύπτει
daughter.acc lead.away.ptc.nom.sg hide.3.sg
'But Demetrios the vinedresser, **having seduced my daughter and taken her away, is hiding her**.'

On a more fundamental level, several treebanking projects for Greek also use different annotation styles. Although the format of the AGDT (based on the Prague Dependency Treebanks) is the most common one, the other major Greek treebank, PROIEL, uses an entirely deviating annotation style. Some other treebanks correspond closely to the AGDT annotation style but have slight deviations. Both the AGDT and PROIEL treebanks have also been converted to Universal Dependencies (UD: see Nivre *et al.* 2016), but even in the UD format there are several structural differences between the two. Additionally, a large collection of Greek text is not available in UD (including not a single papyrus text): altogether the UD treebanks only account for 30% of the total (about 415,000 tokens on a total of more than 1.5 million). Hence, we chose to convert the PROIEL treebanks to the majority format (AGDT), as will be described in Section 4.1.

3.2 Linguistic problems

Even when compared to other inflectional Indo-European languages, Ancient Greek has a notoriously free word order (see e.g. Dik 1995 and Luraghi 2014). In general, free word order languages are more difficult to parse than fixed word order languages, as syntactic relations and dependencies are much less predictable by the linear ordering of words – despite the development of parsers tailored to coping with languages with a free word order in an early stage (cf. e.g. Covington 1990). Additionally, the free word order has several more specific consequences for syntactic parsing. First of all, due to its free word order Greek has an unusually high number of non-projective structures. These are discontinuous constituents, which can be formalised as dependency arcs that connect to the head word while crossing another dependency arc (see, e.g. Osborne 2019). Mambrini and Passarotti (2012), for example, report that one quarter of all arcs used in their experiments to parse Ancient Greek poetry are non-projective, much higher than, for example, Dutch, the language with the highest number of non-projective arcs in the CONLL-X shared task (5.4%: see Buchholz and Marsi 2006). It is well known that non-projective arcs are more difficult to handle than projective arcs for many parsing systems, and many older parsing systems are not even able to predict non-projective arcs (Nivre and Nilsson 2005). Nevertheless, the papyrus corpus shows considerably low rates of non-projectivity in comparison with other Greek texts: while about 41% of all sentences in the full training set we used (mainly consisting of literary Greek: see Section 4.1) include non-projective arcs, this holds true for only about 13% of the papyrus sentences in the training corpus. Second, as Greek does not generally employ

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

word order to mark syntactic relations between constituents, morphological means such as case marking and agreement are typically employed for this end. Consequently, morphology and syntax are highly interrelated, and word forms are of utmost importance to parse the structure of a sentence.

Some specific constructions pose additional difficulties, in particular ellipsis and coordination. In Greek, constituents such as dependents of a predicate or the predicate itself are often left out when inferable from the context or through other means such as agreement patterns: this introduces a high number of 'artificial' elliptic nodes in the data – our training corpus (see Section 4.1) contains 12,970 elliptic tokens on a total of 907,104 (1.4%; see Section 4.2 for more details). Of course ellipsis occurs in other languages as well, but Greek's high rate of ellipsis as compared to languages such as English[e] makes this an important problem to tackle. While it is already difficult to represent coordination – whose nature is symmetric, paratactic and therefore non-hierarchical – in a hierarchical dependency format (see Popel *et al.* 2013: 517, pointing out that symmetric coordination structures tend to be encoded by relying on the same techniques that are used to render asymmetric dependencies, such as edges and nodes), Greek's free word order often entails long distances and discontinuities between the different conjuncts, and the presence of ellipsis may sometimes further complicate the annotation of these constructions. In early experiments, we found that both syntactic coordination and ellipsis lead to a high error rate for Greek, and therefore, these problems are given special attention in Section 4.2.

3.3 The papyrus corpus

The Greek documentary papyrus corpus, which this paper focuses on, is somewhat peculiar for several reasons. First and foremost, many papyrus texts have been transmitted in an incomplete state: due to physical damage to the papyrus, several characters, words or even sometimes complete sentences are regularly missing. This missing text may sometimes be reconstructed on the basis of parallel texts, or simply logical deduction – quite recently, the PYTHIA project (Assael, Sommerschield, and Prag 2019) has also shown exciting possibilities to perform this task in an automatised way, on the basis of machine learning techniques. Nevertheless, in many cases an exact reconstruction remains impossible (e.g. when content rather than function words are missing).

Secondly, the amount of in-domain training material for the papyri is still rather small: the Sematia/PapyGreek treebanks only contained 993 sentences, or 13,018 tokens, and the Pedalion treebanks 2474 sentences, or 29,961 tokens, at the time this paper was written.[f] Consequently, a large share of the training data consists of literary texts (see Section 4.1), while the problem of performance drop in NLP tasks in case of domain shift has been styled 'endemic' in recent research (Elsahar and Gallé 2019: 2163; Ben-David *et al.* 2021: iii). Rather than trying to solve this issue for this paper, we will briefly discuss the impact of genre differences in Sections 5.1 and 5.2.

Finally, the papyri contain several non-standard spellings. However, due to editorial practices, a standardised version of these texts is also available, with a more normalised spelling. These corrections can be broader than spelling normalisations, however: sometimes other linguistic features (e.g. cases) are 'corrected' as well. Since using these non-standard spellings would make syntactic parsing considerably harder – especially since most of the training data is literary, see Section 4.1 – while offering no real advantage (see also Keersmaekers 2020b: 70), we opted to use the standardised version in all cases.

[e] While annotation differences make it difficult to compare treebanking projects cross-linguistically, one indication is the high rate of the 'orphan' relationship in the Greek UD treebanks, which is used in some difficult cases of elliptic relationships (see https://universaldependencies.org/u/dep/orphan.html for more detail). In the Greek PROIEL treebank, this relationship is used 352 times per 100,000 tokens, while it only occurs from 6 to 37 times per 100,000 tokens in the English UD treebanks.

[f] Although PapyGreek has significantly expanded in the meantime, containing 4,012 sentences or 55,134 tokens as of October 2022.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

4. Methodology

4.1 Training, test and development data

Allowing the parser to use as much training data as possible is one method for enhancing parsing accuracy (see Feng *et al.* 2021 for the relevance of data quantity). Our goal was to improve upon past efforts (see Section 2) that used training corpora with fewer than 200K tokens. In order to build a large training corpus, we followed two paths. The first consisted in collecting data from all major Ancient Greek dependency treebanks currently available (see Section 2) and importing them into a relational database, which serves as the back-office of our undertaking.

The annotation styles of both the PROIEL treebank and the Harrington treebank were automatically converted to the AGDT standard on the basis of a rule-based method [the differences between PROIEL and Perseus are described by Celano (2019: 292–93)]. As the PROIEL and Harrington treebanks use a more fine-grained annotation style, this was generally possible without having to rely too much on manual annotation – we followed the guidelines of the respective projects closely to identify the major annotation differences. For PROIEL, the major annotation differences with AGDT involved coordination structures (which have different syntactic labels: see Table 4) and prepositions and conjunctions, which are the head of the noun/verb in the AGDT style but a dependent of the noun/verb in the PROIEL style. The Harrington treebanks do not make a strict distinction between OBJ 'complement' and ADV 'adverbial' in all cases (corresponding to the Prague Dependency Treebank distinction between arguments and adjuncts) – instead, dependents are annotated with the semantic role they express. We therefore introduced this distinction semi-manually and semi-automatically, by distinguishing complements and adverbials based on the lemma of the head. For example, if a word dependent on the verb οἰκέω "live" occurs with the semantic role "location", we can safely assume that it is an "OBJ". We checked the cases in which the degree of argumenthood was not obvious from the lemma-semantic role combination manually. This process was manageable due to the small size of the Harrington treebanks (only 18,000 tokens). We integrated these converted datasets in our database and were able to bring the annotation format closer to the AGDT style. In doing so, we started eliminating inconsistencies and errors, which were caused by a large number of annotators and standards involved, by exploring multiple strategies (from manual punctual corrections to machine learning-based anomaly detection). The effects of such corrections and homogenisations go beyond the level of syntax (part-of-speech, morphological attributes and lemmas are impacted too), and the study of the effects of these interventions requires an extensive and specialised research design. In an ever-growing corpus, which currently already exceeds 1.5 million tokens, such a homogenisation undertaking is doomed to remain work in progress. It would therefore be impossible to claim that our corpus is fully homogenised. Due to the complexity of this issue, we will implement the strategies by for example Dickinson (2015) and analyse their effects in a future, stand-alone paper, in which we will also reflect on underlying theoretical assumptions behind them. In Section 5.2, however, we will further discuss the impact of inconsistencies on the results.

In a second step, we generated new data: relying on the manually annotated treebanks surveyed above, our team (consisting of the two authors of this paper as well as several job students we employed) used the procedure described in this paper to create automatically annotated trees which were afterwards manually corrected (the *Pedalion* trees). Table 1 details the currently available data for Ancient Greek.

The training and development data used for the research underlying this paper are described in Tables 2 and 3. We excluded archaic and classical poetry (e.g. Homer, Sophocles), as well as the late Byzantine Sphrantzes text of PROIEL (15th century AD) from our dataset (405,990 tokens in total, or 28,383 sentences), to avoid too large diachronic and genre differences with the papyri (even though we will evaluate whether excluding the early poetic data was the right choice in Section 5.1.5). As *training* data, we used all of the Sematia/PapyGreek papyri (29% of all papyrus sentences), half of the syntax example sentences, 95% of the classical prose data, 95% of the

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Table 1. Dependency treebanks of Greek available by the end of 2020

Treebanks	Tokens	Texts	Annotation
AGDT (Bamman *et al.* 2009)	c. 560K	Archaic poetry; Classical poetry and prose	Lemmas, morphology, syntax, (semantics)
Gorman (Gorman 2020)	c. 324K	Classical and Post-Classical prose	Lemmas, morphology, syntax
Pedalion (Keersmaekers *et al.* 2019)	c. 300K	Archaic, Classical and Post-Classical poetry andprose	Lemmas, morphology, syntax, semantics
PROIEL (Haug and Jøhndal 2008)	c. 277K	Classical, Post-Classical and Byzantine prose	Lemmas, morphology, syntax, pragmatics
Harrington (Harrington 2018)	c. 18K	Post-Classical prose	Lemmas, morphology, syntax, semantics
Aphthonius (Yordanova 2018)	c. 7K	Post-Classical prose	Lemmas, morphology, syntax, semantics
Sematia (Vierros and Henriksson 2017)	c. 6K	Documentary papyri	Lemmas, morphology, syntax

Table 2. Description of the training data for syntactic parsing

	Sentences	Texts
Papyri	990 (2%)	Letters and petitions from the Sematia papyri
Syntax example sentences	469 (1%)	Easy, short sentences from various authors, used in the modular Pedalion grammar (Van Hal and Anné, 2017)
Classical Prose	15,863 (33%)	Herodotus (34%), Xenophon (15%), Demosthenes (11%), Thucydides (8%), Plato (7%), Lysias (6%), Aristoteles(5%), Antiphon, Aeneas Tacticus, Aeschines, Isocrates, Hippocrates
Postclassical Prose	28,744 (60%)	New Testament (36%), Polybius (12%), Athenaeus(8%), Septuagint, Procopius, Dionysius of Halicarnassus, Plutarch, Flavius Josephus, Diodorus of Sicily, Appian,Life of Aesop, Aesop, Lucian, Sextus Empiricus, Pseudo-Lucian, Epictetus, Theophrastus, Aphthonius, Paeanius,Chion's Letters, Phlegon, Epicurus, Pseudo-Apollodorus, Julian the Emperor, Longus, Nicene Creed
Postclassical Poetry	1499 (3%)	Menander (61%), Herodas (27%), Batrachomyomachia (10%), Theocritus (1%)

postclassical poetry data and 90% of the postclassical prose data, together amounting to 47,565 sentences (907,104 tokens). For the *development* data, we wanted to stick as closely as possible to the target domain with regard to genre and diachrony, including only data from papyri (30% of all sentences in the Pedalion treebanks), a quarter of the syntax example sentences, 5% of the postclassical prose data and a short inscription. Finally, this left us with 70% of the sentences in the Leuven papyri as *test* data (1677 sentences, or 20,869 tokens). These texts are nearly all letters (96% – the other 4% are petitions), while 19% are from the Hellenistic period (3rd–1st century BC), 77% from the Roman period (1st–4th century AD) and 4% from the Byzantine period (4th–8th century AD). Section 5.1.12 will also compare the results of literary texts to the papyri for comparison. For this purpose we used all remaining literary data (i.e. 25% of the example sentences and 5% of classical and post-classical prose as well as postclassical poetry – 2746 sentences, or 51,538 tokens in total).

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Table 3. Description of the development data for syntactic parsing

	Sentences	Texts
Papyri	718 (28%)	Letters and petitions from the Leuven papyri
Inscriptions	15 (1%)	Section from the Parian marble
Syntax example sentences	234 (9%)	Same as training data
Postclassical Prose	1597 (62%)	Same as training data

4.2 Manipulating the data

Given that this article pays special attention to the impact of the annotation format on the achievements of the parser, this section describes a series of rule-based transformations we employed to tackle three particularly pervasive problems: coordination and ellipsis (see Section 3.2) as well as textual damage (see Section 3.3).

First of all, the Greek treebank data encodes **elliptic structures** with dummy nodes or empty heads, that is extra tokens outside of the sentence indicating an elliptic word that should be inserted (see the left side of Figure 1 for an example). These nodes had to be removed in some way, as the parsers we tested were not able to insert new tokens in the test data. We chose to encode these elliptic tokens by relying on their syntactic labels, using a "composite" presentation. This principle is illustrated in Figure 1, where the word with form *[0]* refers to an elliptic verb, in compliance with the AGDT guidelines. In the transformation implemented, this *[0]* is deleted, while another token is assigned a composite relation. The label "PRED/SBJ", for example, means that the token is in a subject relationship with an elliptic predicate. The advantage of using this representation format is that it is easy to reconstruct the original elliptic tokens (e.g. when a word has the PRED/SBJ relationship, we can add an elliptic "PRED" node and attach this word to it with the "SBJ" relationship).

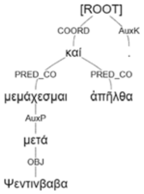
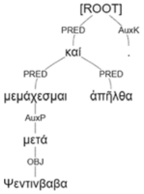
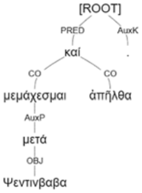
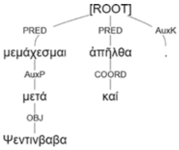
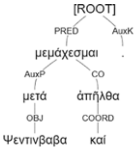
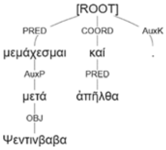
The drawback is that this method heavily proliferates the number of relations that the parser has to take into account. Therefore, we created an additional version of the data in which three highly frequent sources of ellipsis were further reduced: elliptic copula, comparative constructions and infinitives without a main verb. These transformations are illustrated in Figures 2–4. Although they only account for about half of all elliptic structures (49% of all elliptic nodes in the test data), this allows us to gain a first estimate on how impactful reducing elliptic structures would be. We will evaluate the impact of this method in Section 5.1.8.

Secondly, the problematic status of **coordination** has already been described in detail by Popel *et al.* (2013), who analyse a number of different representational formats for coordination structures in syntactic dependencies, some of which might be easier to handle for a syntactic parser. Therefore we tested a number of alternative representation formats for these structures and their impact in some early experiments with *MaltParser* (Nivre *et al.* 2007) on a test corpus containing 162 coordination structures in the papyri. These formats are summarised in Table 4. While coordination styles 1 and 2 in Table 4, which are used by the major Greek treebanks and which share the fact that the coordinator serves as head and that the group is kept intact, returned a similar LAS for coordination constructions specifically (0.45), parsing accuracy for these constructions considerably increased from this number when using either encoding strategy 4 or 5, where conjuncts serve as head (to 0.74 and 0.72 respectively). By adopting strategy 3 and 6, we saw a moderate increase (0.52 and 0.62 respectively).[g] Since strategy 5 (which, coincidentally, is used by the UD treebanks) kept the coordination group intact, so that no syntactic information was

[g] The sample was 463 coordination structures. The increases in LAS of strategy 3, 4, 5 and 6 over strategy 1 were all statistically significant, with p<0.01 with McNemar's test.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Table 4. Select possible representational formats for coordination structures, illustrated through the sentence μεμάχεσμαι "I fought" μετὰ Ψεντινβαβα "with Psentinbaba" καὶ "and" ἀπῆλθα "I went away"

Strategy	Example	Used in	Head	Relation of group	Group intact
1	 [ROOT] COORD AuxK καί . PRED_CO PRED_CO μεμάχεσμαι ἀπῆλθα AuxP μετά OBJ Ψεντινβαβα	AGDT, Prague DependencyTreebanks	Coordinator	Conjuncts (_CO)	YES
2	 [ROOT] PRED AuxK καί . PRED PRED μεμάχεσμαι ἀπῆλθα AuxP μετά OBJ Ψεντινβαβα	PROIEL	Coordinator	Conjuncts + Coordinator	YES
3	 [ROOT] PRED AuxK καί . CO CO μεμάχεσμαι ἀπῆλθα AuxP μετά OBJ Ψεντινβαβα	/	Coordinator	Coordinator	YES
4	 [ROOT] PRED PRED AuxK μεμάχεσμαι ἀπῆλθα . AuxP COORD μετά καί OBJ Ψεντινβαβα	/	Conjuncts	Conjuncts	NO
5	 [ROOT] PRED AuxK μεμάχεσμαι . AuxP CO μετά ἀπῆλθα OBJ COORD Ψεντινβαβα καί	UD	First conjunct	First conjunct	YES
6	 [ROOT] PRED COORD AuxK μεμάχεσμαι καί . AuxP PRED μετά ἀπῆλθα OBJ Ψεντινβαβα	/	Coordinator	Conjuncts	NO

lost (i.e. which words are coordinating with which other words), we have stuck to this strategy: its impact on the parsing data is discussed in Section 5.1.7. In this strategy, we attached shared modifiers among several conjuncts (e.g. "**the** men and women", where "the" applies to both "men" and "women") to the first conjunct (as is the case in UD), unlike the AGDT style, which attaches shared modifiers to the coordinating conjunction. As this has implications for the conversion back to the AGDT style, we will further discuss the impact of our conversion of coordination constructions in Section 4.4.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

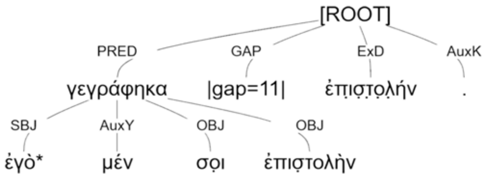


Figure 5. Representational format for damaged sentences: “ ἐγὸ μὲν σοι ἐπιστολὴν γεγράφηκα [gap of 11 characters] ἐπιστολήν” “I have written you a letter . . . letter”. The asterisk refers to the original spelling.

Finally, as mentioned in Section 3.3, the papyrus data also contain several gaps. For the time being, we decided to replace these sections with a dummy token “GAP”, and still annotated the parts of which the internal relations are clear, while any words of which the syntactic relationship is unclear were simply attached to the root with the relation “ExD” (used in the AGDT for all extra-clausal constituents, e.g. vocatives, parenthetical clauses). This is illustrated in Figure 5: it is still possible to annotate the internal syntactic structure of “ ἐγὸ μὲν σοι ἐπιστολήν γεγράφηκα” “I have written you a letter”, whereas the second mention of ἐπιστολήν “letter”, following the long gap, is annotated as an external constituent. While in the future a more elegant method is necessary to deal with this problem (e.g. incomplete parsing, see Vilares, Darriba, and Vilares 2004 for a solution proposed in constituency parsing), the current method proved sufficient to achieve an acceptable accuracy, as will be shown in Section 5.2.

4.3 Training parsers

In our initial experiments, like Mambrini and Passarotti (2012) (see Section 2) we made use of MaltParser (Nivre *et al.* 2007), a library of transition-based parsing systems making use of either support vector machine or linear classifiers on a set of manually defined features, with a variety of deterministic parsing algorithms implemented. While a relatively old parser, one of its advantages is that is highly configurable and allows for automatic feature optimisation through *MaltOptimizer* (Ballesteros and Nivre 2012). However, relying on a pilot study carried out by Mercelis (2019), we found a strong increase in parsing accuracy with Stanford’s Graph-Based Neural Dependency Parser (Dozat, Qi, and Manning 2017), one of the top-scoring parsing systems for inflectional languages in the 2017 and 2018 CONLL shared tasks on multilingual parsing. More precisely, on a dataset of 38,427 test tokens of literary Greek, LAS increased from 0.754 (MaltParser) to 0.797 (Stanford). The LAS for Ancient Greek ranged from 0.732 (for the Perseus treebanks) to 0.743 (for the PROIEL treebanks) (Dozat *et al.* 2017) in earlier tests. Stanford’s parser is a graph-based parsing system that represents words as the sum of (a) pretrained word embeddings, (b) an embedding of the word token, (c) a character-level embedding of the word form and (d) embeddings of part-of-speech and extended part-of-speech tag, the latter three of which are produced by an LSTM (long short-term memory, a type of artificial neural network i.e. able to learn long-term dependencies) network (see Dozat *et al.* 2017 for more details). There are multiple reasons why this system performs well for Greek and other inflectional languages: through its character-level word embedding, it is able to deal with languages with complex morphology, while its graph-based system is also able to handle non-projectivity (see Section 3.2) in a satisfying way: as shown by Dozat *et al.* (2017: 25–26), performance significantly decreases with non-projective arcs using a transition-based parsing system as compared to a graph-based system.

The input for the parser is a CONLL file, of which the relevant columns for the parser are summarised in Table 5 – while the parser does not take morphological information (‘Features’) into account, we will discuss the impact of using morphology rather than coarse-grained part-of-speech tags in Section 5.1 below. As for the other columns, the part-of-speech tag is a very broad, five-way classification between nouns, verbs, adjectives, function words and punctuation marks.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Table 5. Example of a Greek sentence in CONLL format (without Lemma; sentence: πάλιν καθεύδεις; “are you sleeping again?“)

ID	Form	POS	XPOS	Features	Head	Relation
1	πάλιν	Function word	Adverb	_	2	ADV
2	καθεύδεις	Verb	Finite	Person = 2; number = sg; tense = pres; mood = ind; voice = act	0	PRED
3	;	PUNCT	PUNCT	_	1	PUNCT

The extended part-of-speech is more fine-grained (for nouns: common noun, proper noun, personal and relative pronoun; for verbs: finite, infinitive, participle; for adjectives: adjective proper, article and numeral; for function words: adverb, coordinating conjunction, subordinating conjunction, preposition, particle and interjection). The head and relation columns follow the Perseus annotation style, with some extra labels based on the transformations we conducted on the data (see Section 4.2).

4.4 Converting the data back to its original format

In Section 4.2, we described a number of manipulations to make the data more easily ‘learnable’ for a parser (see Section 5.1 for an evaluation of the results) or to handle specific challenges in the data that needed to be tackled in some way (i.e. elliptic constructions and damaged sentences). For several purposes, however, for example, when creating new treebanks that are consistent with the AGDT format (see Section 6 for more detail), it is necessary to retrieve back the syntactic structures as they are defined in the AGDT guidelines rather than the idiosyncratic annotation format we used for coordination and elliptic structures. For this purpose, we wrote a rule-based procedure to convert the data back to its original format. This involved (a) defining the conjunction and conjunct of coordination structures and attaching the conjuncts back to the conjunction; (b) generating elliptic nodes when a word is in a “composite” elliptic relationship and attaching the word to this new elliptic node and (c) inserting additional elliptic nodes when an elliptic construction has been simplified (i.e. a conversion back in the opposite side of the arrow of Figures 1–4, and a conversion back from strategy 5 in Table 4 to strategy 1).

To evaluate whether the procedure we developed was working correctly, we tested our back-conversion procedure on the training data after first exercising our manipulations on it. We then compared how close the “back-converted” training data was to the original training data by calculating the overlap between heads and relations of both datasets. This analysis showed that the conversion back procedure was certainly not perfect: there was only a 94% match between the heads of the original training data and the “back-converted” training data. Some errors were simply related to complex cases that our back-conversion code did not fully cover, or because of inconsistencies in the original training data (e.g. we tried to restore elliptic verbs for comparative constructions – see Figure 3 – but there were some cases of comparative constructions in our training data that did not have a elliptic verb to start with although they should have). However, there were a number of constructions where the conversion led to some information loss.

First, as mentioned in Section 4.2, we generally encoded elliptic structures with composite relations, for example ‘ADV/ATR’ for an attribute of an elliptic adverbial constituent. For the back conversion, one could then generate an elliptic adverbial node with the relation ‘ADV’ and attach the original ‘ADV/ATR’ node to it with the relation ‘ATR’. However, there is a problem when there are multiple nodes with, for example the relation ‘ADV/ATR’: this could either mean that there are two elliptic ‘ADV’ nodes with an ‘ATR’ node attached to each of them or one elliptic ‘ADV’ node with two elliptic ‘ATR’ nodes attached to them. To ensure that the number of elliptic nodes would not proliferate, we simply generated one elliptic node in such a case, even though this means that the conversion fails in some cases where there should be more than one elliptic node. This problem is generally inevitable: as the test data would not contain any elliptic nodes, it is necessary to find a way to deal with this nodes (although an alternative method is to use a parsing technique that can automatically generate elliptic nodes when necessary: see e.g. Seeker *et al.* 2012).

Second, as mentioned in Section 4.2, shared modifiers of multiple conjuncts were attached to the first conjunct. However, this means that modifiers of the first conjunct are always ambiguous: they could either be modifiers shared by every conjunct or only modifiers of the first conjunct. Therefore, the conversion in coordination style leads to a certain amount of information loss. However, when we performed a test with coordination structures encoded in the Perseus format (see Section 5.1.7), shared modifiers had a very low parsing accuracy: only 30.7% of the shared modifiers were attached to their correct head. As our coordination format was able to increase parsing accuracy for coordination structures in general to a large extent (see Section 5.1.7), one could argue that this amount of information loss is acceptable for a structure that the parser would struggle with anyway. Admittedly, there are other ways to handle shared modifiers in the annotation format we are using: they could be attached to one of the conjunctions instead of the first conjunct, or they could be assigned to multiple heads together with a parsing method that can handle multi-headedness.

5. Results

This section assesses the impact of the methodology described in Section 4. Section 5.1 gives a broad overview of the results of the different strategies we applied, measured by typical evaluation metrics. Section 5.2 offers a detailed qualitative analysis of the remaining errors and suggests rooms for further improvement.

5.1 Model comparison

5.1.1 Introduction

We will here discuss the impact of the strategies outlined in Section 4 by making use of the most popular metrics for assessing dependency parsers. The Attachment Score refers to the percentage of words that are attached to the correct head. In the Unlabeled Attachment Score (UAS), the relationship label is ignored. The percentage of words which is given both the correct syntactic head and relationship label is known as the Labeled Attachment Score (LAS). Conversely, the Label Accuracy (LA) refers to the proportion of tokens whose syntactic relations are correctly predicted, irrespective of the predicted heads (for more information on these and alternative metrics, see Kübler *et al.* 2009: 79–80).

If we simply let the parser work based on the training data presented above without performing any further manipulations (this is the 'Base' model), we obtain an LAS score of 0.793 (cf. Table 6). We first performed a number of tests changing only one parameter in comparison with the base model (Section 5.1.2–5.1.9) and then combined several parameters in one model (Section 5.1.10–5.1.11). While the parser also predicts a head and label for punctuation tokens and tokens indicating damage to the text, their head and relation can be predicted automatically in almost all cases, as they are typically attached to the root with the label "PUNCT" (our own homogenisation of the labels 'AuxK', 'AuxX' and 'AuxG' used for punctuation in the original treebanks) or "GAP", respectively. So as to not overinflate parsing accuracy, we therefore excluded them from the evaluation. In what follows, we will discuss each of these operations one by one. Table 6 shows the impact of the various strategies applied on the Greek papyri.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Table 6. Overview of the main results of the test set (papyrus corpus, with $N = 17{,}609$)

Model	UAS	LA	LAS
Base	0.848	0.849	0.793
Unicode encoding	0.843 (−0.5%)	0.844 (−0.5%)	0.784 (−0.9%)
Accents removed	0.841 (−0.7%)	0.840 (−0.9%)	0.780 (−1.3%)
PROIEL data removed	0.817 (−3.1%)	0.819 (−3.0%)	0.749 (−4.3%)
Poetic data included	0.843 (−0.5%)	0.846 (−0.3%)	0.788 (−0.5%)
SVD word vectors	0.848 (+0.0%)	0.850 (+0.1%)	0.793 (+0.0%)
Improved coordination	0.877 (+2.9%)	0.871 (+2.2%)	0.825 (+3.2%)
Ellipsis reduced	0.862 (+1.4%)	0.872 (+2.3%)	0.808 (+1.5%)
Morphology included	0.859 (+1.1%)	0.854 (+0.5%)	0.795 (+0.2%)
Combined (vectors + coordination + ellipsis + morphology)	0.898 (+5.0%)	0.900 (+5.1%)	0.851 (+5.8%,)
Combined + automatically predicted morphology	0.894 (+4.6%)	0.894 (+4.5%)	0.845 (+5.2%)

5.1.2 Beta code versus unicode

Due to technical restrictions of the tagger used in a preliminary phase in the pipeline, the Greek data for morphological analysis were converted into *Beta Code*, in which Greek letters are encoded by Latin characters, and diacritical marks such as accents are represented by adding symbols next to these Latin characters (see Verbrugghe 1999): for example the Greek character έ is represented as *e/* in Beta Code (see also Table 5). A test in which all data were entered in composed Unicode format (i.e. e/ corresponds to just one character έ) yielded unequivocally negative results: an overall LAS reduction of 0.009 (which is statistically significant with McNemar's test, $p < 0.0001$). It is therefore our hypothesis that the compositional nature of Beta Code provides the parser with useful information (of course, a similar effect can be achieved by using decomposed Unicode characters, in which έ corresponds to two characters). As 'complex' characters such as έ are decomposed into two characters (*e* and */*) the character set is considerably shorter (the number of characters is reduced from 144 to 62). Consequently, this suggests that other languages with a similarly high number of diacritical marks may also benefit from such a 'decomposition'.

5.1.3 Removal of accents

Ancient Greek is one of the few languages in which the word accent is indicated in writing, which results from a mere convention among editors. For this, three different diacritical signs are used, which have already been reduced to two in our back-office environment (the so-called gravis accent only appears on the last syllable as a conventional variant of the so-called acutus, and we have therefore replaced it by the acutus in all cases). It was our hypothesis that the presence of the accent would overall add little information for the parser. However, when we trained a model without accents, LAS decreased with 0.013 ($p < 0.0001$ with McNemar's test). In other words, accents appear to provide useful disambiguating information to the parser.

5.1.4 Removal of the PROIEL data

One of the corpora we included in our training set, the PROIEL corpus, was automatically converted into a different annotation format, as described in Section 4.1. As this conversion, which was as such not impeccable, may introduce additional errors to the parser, we evaluated the impact

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

when these data were removed from the training/development set. This clearly has a strong negative effect: the LAS drops with 0.043 ($p < 0.0001$ with McNemar's test), suggesting that the quantity of this dataset (243,951 tokens) is more important than the fact that it still includes several conversion-related mistakes. Additionally, the New Testament part of the PROIEL corpus may be especially suitable training material for the papyri, as it is diachronically situated in the same period and does not use a too elaborate syntax (as opposed to, e.g. philosophical or poetic texts).

5.1.5 Inclusion of early poetic data

In our experiments, we excluded the early poetic data from the training corpus, as we considered them to be too far removed in stylistic and diachronic terms from our test corpus to be useful as training material (for some late papyri, the distance in time between the Homeric data is as big as it is with modern Greek data, e.g.). As one could wonder whether this was the right choice, we performed an additional experiment with the early poetic data included in the training corpus. This experiment confirms our hypothesis: parsing accuracy was slightly lower with the early poetic data included, with a drop in LAS of 0.005 ($p = 0.016$ with McNemar's test).

5.1.6 Use of pretrained SVD embeddings

While we used the pretrained word embeddings for Greek created by the Language Technology Group of the University of Oslo[h] in our other tests, we also wanted to evaluate the impact of our own pretrained embeddings, which incorporate syntactic information and therefore might be particularly suitable for syntactic parsing – (Keersmaekers 2020a) showed that they were also quite successful in another NLP task, viz. semantic role labelling. These vectors were created by singular value decomposition (SVD) on a matrix of PMI associations between a word and its syntactic dependents (see Keersmaekers 2020c for more detail). As they were created on the basis of word lemmas rather than forms (as required by the Stanford parser), we transformed them to form vectors by simply assigning the same vector to each form of a particular lemma. If a form could have different lemmas, we assigned the weighted average vector of these lemmas to the specific form based on the frequencies of each possible lemma (e.g. if form X was analysed two times as lemma Y and fourtimes as lemma Z, lemma Z would be weighted double in the vector of form X). This method, however, had unsatisfactory results, showing almost no increase in LAS (+0.0006, $p = 0.785$ with McNemar's test). Perhaps directly calculating pretrained embeddings on the basis of the word form, or, alternatively, making the parser directly use lemma rather than form embeddings in its parsing model, would increase the results. This latter strategy seems preferable for highly inflectional languages such as Greek, as the high number of possible word forms for a given lemma would quickly lead to data sparsity issues (although perhaps this could be resolved by using subword-based embeddings such as FastText: we leave the question whether subwords can accurately represent Greek morphology for further research).

5.1.7 Transforming coordination structures

Section 4.2 described possible ways to transform coordination structures so that they would be easier to 'interpret' for the parser. When one of these encoding strategies is adopted (more precisely, strategy 5 in Table 4), there is a strong improvement in automatic parsing quality, as shown by the numbers in Table 6: the LAS increases with 0.032 (to 0.825, $p < 0.0001$ with McNemar's test), mainly through improved head attachment (UAS + 0.029) but also improved labeling (LA + 0.022), likely because the number of syntactic relations is significantly reduced (due to the removal of '_CO'-suffixes, see Section 4.2). This effect is even stronger if we only consider

[h] Created by a Word2Vec SkipGram model on the Ancient Greek treebanks, see http://vectors.nlpl.eu/repository/.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

words that are part of a coordination structure (defined as words with a coordinating conjunction as its head or the coordinating conjunction itself): the LAS for these words raises from a meagre 0.684 to 0.833, that is even a little better than the average parsing accuracy for all words (0.825).

5.1.8 Reducing ellipsis

As described in Section 4.2, ellipsis is encoded in such a way that it heavily proliferates the number of syntactic relations in our data. We therefore used several rules to significantly reduce the number of elliptic structures (almost halving them in the test data). As can be inferred from Table 6, this has a considerable impact on the results (LAS + 0.015, $p<0.0001$ with MCNemar's test). However, this is for a large extent caused by a technical issue: the Stanford neural parser required its output to be a tree with only one node at its top (i.e. only one word without head). While we ensured that the data we used initially followed this principle, after removing elliptic nodes there would sometimes be trees with multiple nodes at its top (if the node at the top of the tree was elliptic), as illustrated in Figure 1, so that these trees could never be parsed correctly. Several of the ellipsis-removal strategies we used were able to avoid this issue, thus boosting parsing accuracy, as illustrated in Figures 2 and 4. However, in sentences where this issue did not arise (i.e. with only one node at the top of the tree), parsing accuracy did not improve (the LAS was 0.824 for these sentences, with $N=2185$, while it was 0.826 in the model where ellipsis was not reduced). As we were only able to remove about half of all elliptic structures, this still left us with a large set of special 'elliptic' relations: perhaps this issue can only be avoided by fully exterminating elliptic constructions from the data. An alternative hypothesis is that the parser was in fact able to handle the encoding strategy we used for elliptic structures (as described in Section 4.2) quite well: however, looking at the parsing results for tokens depending on an elliptic node specifically, this was clearly not the case – the LA for these tokens was only 0.379 in the base model ($N=966$), as opposed to the average LA of 0.849.

5.1.9 Adding morphological features

Morphology (e.g. case usage) is highly important in syntax for an inflectional language such as Greek. However, the Stanford Neural Parser does not normally take morphological features into account, as it only builds embeddings on the basis of the 'FORM', 'POS' and 'XPOS' columns of the CONLL input (see Table 5). Hence, we tested the impact of introducing such morphological information to the parser by putting it in the place of the coarse-grained POS tags (which might be too broad to present useful information to the parser to start with). Although this column has a large number of possible values (656 possible combinations of morphological features), adding it has a small positive impact on parsing accuracy, with LAS + 0.002, UAS + 0.011, and LA + 0.005. The fact that the Stanford parser builds embeddings based on word characters, so that morphology is already represented in some way in the parser's model, may explain why this impact is not significantly higher. Nevertheless, this character-based model is likely not able to capture all relevant information with regard to Greek morphology, as shown by the increase in accuracy if morphological features are added (although the improvement on LAS specifically is rather small, and this improvement is not statistically significant with $p=0.26$ with McNemar's test, so a chance result cannot entirely be dismissed as a possible explanation). A future strategy could consist in selecting a restricted number of features, for example by excluding person and aspect/tense.

5.1.10 Combining multiple strategies

In a next step, we combined the strategies that had a positive or a neutral effect (the new coordination structures, the addition of our own SVD vectors, the reduction of elliptic structures and the addition of morphological features). The cumulative LAS score is 0.851, an improvement of 0.058

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Table 7. Results of combined model with automatically predicted POS and features (gold morphology)

Training/validation data	Test data	LAS
Gold	Gold	0.851
Automatically predicted	Automatically predicted	0.845
Gold	Automatically predicted	0.838

($p < 0.0001$ McNemar's Test), higher than the sum of the gains in the individual tests (0.049). This seems to suggest that these strategies fruitfully reinforce each other.

5.1.11 Using automatically predicted part-of-speech and morphology

The tests described above make use of gold morphology in the test data, which does not show how the parser would perform in a 'real-life' application, where unannotated data enter a pipeline. Hence, we performed an additional test, based on the combinatory model, in which the part-of-speech and morphology was automatically predicted (the method is described in Keersmaekers 2020b). We predicted the morphology for the test and development data on the basis of an *RFTagger* model (Schmid and Laws 2008) trained on the training data, while we divided the training data in ten folds and used the rest of the data each time to train ten models to automatically predict morphology/POS for each part of the training data as well. This last step allowed the parser to handle noisy (i.e. not completely accurate) morphology in its training model. Luckily, the impact of automatically predicted morphology and part-of-speech is only minimal: UA drops by 0.004, while both LA and LAS drop by 0.006, resulting in a final LAS of 0.845. Using automatically predicted morphology/POS is clearly beneficial: when the parser was trained with gold morphology and POS in the training and development data (see Table 7), LAS would drop to 0.838, more than double the decrease as compared to automatically predicted morphology (the difference between the model with gold and automatically predicted training data is also statistically significant, $p < 0.0001$ with McNemar's test).

5.1.12 Parsing accuracy for literary texts

Finally, we compared the results of the papyri to literary texts, using the model with automatically predicted morphology. The results, grouped by genre, are summarised in Table 8: as the results for genre were sometimes influenced by one specific author who was overrepresented in the genre in question (e.g. 2 of the 10 authors in the historical genre, Herodotus and Polybius, constitute about half of all tokens for this genre), we also include calculations for the median author within each genre and the standard deviation (in this case we only consider authors with at least 100 test tokens). As there were no texts that had remarkable differences between UAS and LA, we simply report the LAS for each genre. The authors included are the same as the training data, as summarised in Table 2 (see Section 4.1). By means of comparison, the result for the papyri is also included.

The syntax example sentences, as part of an online grammar, were specifically chosen to be simple and easily interpretable (their average length is only 8.3 words, as compared to 16.4 for the full dataset), so their high parsing accuracy is to be expected. A manual inspection of the data also revealed that several of the 'mistakes' the parser found were instead annotation errors in the gold data. Even more accurately parsed are religious texts (the Septuagint and the New Testament). Given that their language is relatively plain, they also use slightly shorter sentences on average (14.6 words), they contain several formulaic constructions and there is a large number of religious training material (see Table 2 – given that the proportion of literary genres is the same as in the

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Table 8. Results of model with automatically predicted morphology by genre

Genre	*N* tokens	Mean LAS	Median LAS	Std dev
Papyri	17,609	0.845	–	–
Religion	8166	0.881	0.873	0.010
Syntax example sentences	1637	0.870	–	–
Biography	1445	0.832	0.832	0.001
Epistolography	255	0.828	0.803	0.037
History	14,393	0.825	0.825	0.029
Oratory	4482	0.822	0.818	0.025
Narrative	2019	0.804	0.820	0.066
Dialogue (Non-Philosophical)	2329	0.798	0.782	0.025
Philosophical dialogue	2431	0.790	0.790	0.040
Philosophy & Science	1608	0.751	0.758	0.024
Poetry	622	0.740	0.726	0.058

training data, the numbers in Table 8 are representative as well), this can also easily be explained. Five other prose genres – biography, epistolography, history, oratory and narrative prose – show similar LAS scores: about one to two percentage points lower than the LAS of the papyri. While the mean is a little lower for narrative prose, this is mostly caused by narrative texts of Lucian (primarily the *True Histories*), with an LAS of only 0.664 ($N = 256$ tokens). The *True Histories* text was part of the Harrington Treebanks, however, which involved an automatic conversion to the Perseus annotation style (see Section 4.1), so problems in the conversion process may possibly explain its low parsing accuracy. Other than that, the frequent use of direct speech in narrative texts may also be difficult to handle for the parser.

The other four genres are stylistically further removed from the rest of the corpus, and accordingly they also have a lower parsing accuracy. Dialogues include a large number of particles and direct quotations (additionally, the dialogues of Athenaeus also have several quotations from poetic texts such as Homer), which may be difficult for the parser to handle. The abstract subject matter for philosophical dialogues presents additional difficulties (e.g. unusual constructions such as nominalisations and neuter subjects). This also explains why other philosophical and scientific texts are hard to parse. Finally, it is unsurprising that poetic texts have a low accuracy rate, since 97% of the training data and all of the development data consisted of prose (see Section 4.1). A training set that includes more poetic material would likely improve parsing. Nevertheless, some poetic texts performed better than others – in particular the *Batrachomyomachia* (an epic parody), with an LAS of 0.813 ($N = 96$), and Menander's *Dyskolos* (a comedy), with an LAS of 0.784 ($N = 269$). Although these texts have significant drawbacks for the parser (an even more free word order, more dialectal forms), they also have shorter sentences (only 9.9 words on average), so a more dedicated approach to Greek poetry would likely be able to significantly improve on this first result.

5.2 Detailed error analysis

While the metrics used in the previous section are helpful to gain an overview of parsing quality, they do not provide a full picture. Plank *et al.* (2015) show, for example, that while LAS is

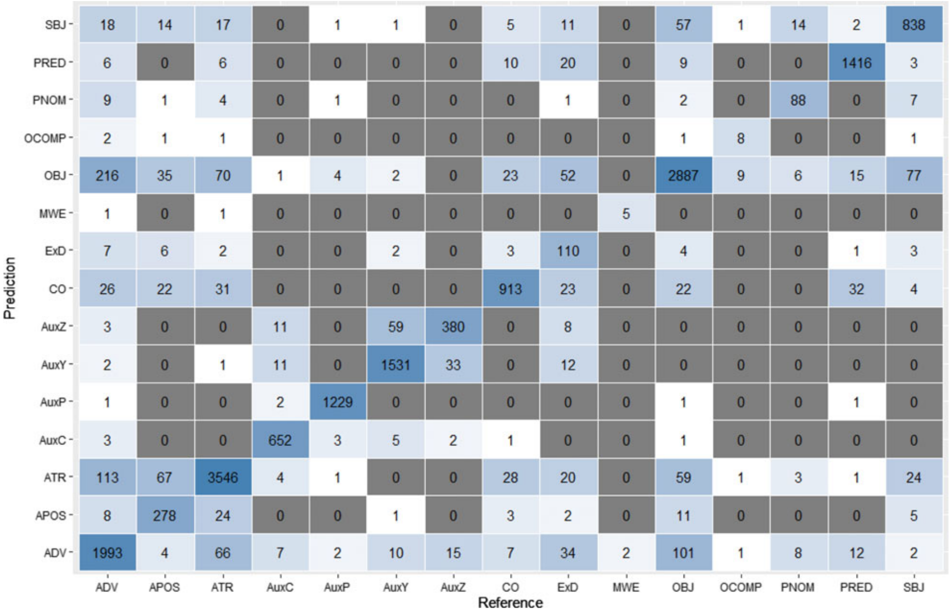


Figure 6. Confusion matrix of syntactic relations (colours are normalised by total counts).

the parsing metric that correlates most strongly with human judgements, the correlation is still relatively weak, and it fails to capture certain human preferences – for example a stronger importance attached to syntactic heads than to relations and preference of content over function words. For our data, several inconsistencies in both training and test data may also distort the numbers (see below). This section therefore provides a more detailed error analysis, starting from the best performing 'realistic' model (i.e. with automatically analysed morphology, see Section 5.1.11).

Figure 6 is a confusion matrix detailing which syntactic relations are most frequently confused with which other syntactic relations. For reasons of simplicity, we included 'composite' elliptic relations such as 'PRED/ADV' (see Section 4.2) as the relation of the non-elliptic word, in this case 'ADV'. The syntactic relations are the following: ADV (adverbial), APOS (apposition), ATR (attribute), AuxC (conjunction), AuxP (preposition), AuxY (sentence-level particle), AuxZ (word-level particle), CO (conjunct), ExD (sentence-external constituent), MWE (multi-word expression), OBJ (complement), OCOMP (object complement), PNOM (predicate nominal), PRED (predicate) and SBJ (subject).

One very frequent error is the use of the label 'OBJ' (complement) instead of 'ADV' (adverbial) (and to a lesser extent, 'ADV' instead of 'OBJ'): this will be further discussed below. Moreover, adverbials ('ADV') sometimes get confused with attributes ('ATR'): this happens relatively frequently with genitive temporal expressions, which are rather peculiar to the papyri, so an expanded set of papyrus training data will likely help to resolve this issue.[i] Appositions frequently get confused with attributes, likely due to inconsistencies in both training and test data (see below). The identification of attributes is relatively unproblematic (with an accuracy of 0.94), just like function words such as conjunctions (AuxC: 0.95), prepositions (AuxP: 0.99), sentence-level particles (AuxY: 0.95) and word-level particles (AuxZ: 0.88), although the latter two occasionally get confused with each other, likely due to inconsistencies in the

[i] These are expressions such as ἔτους δ Μεσορὴ β "[written] **in the** fourth **year**, the second of Mesore (an Egyptian month name)". Besides the fact that the genitive case is rather infrequently used for adverbial groups, these expressions also typically involve an elliptic verb, further complicating their automated processing.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Table 9. Qualitative analysis of 500 parsing errors

Error type	Frequency
Grammatical mistake	277/500 (55%)
Consistency issue	130/500 (26%)
Annotation error in the test data	48/500 (10%)
Technical problem (multiple nodes without head)	20/500 (4%)
Damaged text	13/500 (3%)
Ambiguous sentence structure	12/500 (2%)

training/test data (see below). Conjuncts are also labelled relatively accurately (with an accuracy of 0.92), thanks to the adoption of a new annotation format for coordination (see Section 5.1.7). The labelling of external constituents (ExD) is more problematic, with an accuracy rate of only about 0.38. This is caused by a high proportion of parenthetical verbs which were often labelled as the main predicate, while the main predicate was attached as a conjunct (e.g. ἐρωτῶ σε μεγαλῶς καὶ παρακαλῶ, ἐπιμέλου ἑαυτῆς "I strongly **ask** and **beg** you, take care of yourself)". As the number of verbs that occur in such parenthetical constructions is rather limited, this issue may be resolved with rule-based post-processing. Other than that, the label *ExD* is also used in cases of textual damage (see Figure 5 in Section 4.3), and these sentences are obviously quite tricky to parse (see also below). The relations 'MWE' (multi-word expression) and 'OCOMP' (object complement) are too infrequent to say anything about their prediction accuracy. The label 'OBJ' is predicted relatively accurately (0.92), as it is most of the times rather straightforwardly expressed by the accusative, although it occasionally gets confused with an adverbial, as will be discussed below. Predicate nominals ('PNOM') occasionally get confused with subjects, typically in cases of copula verbs with only one nominative. The main predicate is identified correctly in the vast majority of cases (0.96). Finally, subjects also have a relatively high accuracy (0.87), although they are confused with complements in a significant number of cases (8%). Most of these cases either involve neuter subjects (which do not express any formal difference in the nominative and accusative case) or the subjects of non-finite verbs, which are always expressed in the accusative. In both cases, better quality semantic information (e.g. pretrained word embeddings) may help, as there are many cases in which certain participants (e.g. animate or inanimate participants) are much more likely to be the subject or object, depending on the semantics of the main verb.

Next, we performed a qualitative analysis of a sample of 500 parser errors (i.e. the first 500 in our test set) for the same parsing model. The result of this analysis is summarised in Table 9.

A first striking fact is that truly syntactic parsing mistakes only account for about half (55%) of all errors. A significant group of problems, about one quarter (26%), proves to result from consistency issues.[j] In these cases, the automated analysis turned out to be deviating from the gold data due to inconsistencies in the training data, rather than being wrong. These problems are relatively diverse: the most frequent ones include the attachment of particles (16/130, see sentence [2] for an example), the head word of appositive phrases (15/130, as in [3]), articles that follow their noun instead of preceding them (9/130, as in [4]) and the syntactic head, when a word is dependent on a verb + infinitive complement (8/130, as in [5]), although there is a wide range of other consistency issues (in total, we counted thirty-three categories of inconsistencies).

[j] Note that Bamman et al. (2009) report an interannotator agreement of 0.874 for syntactic heads, 0.853 for syntactic relations and 0.806 for both for the Ancient Greek Dependency Treebank, so some level of inconsistent annotation is expected.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

(2) εἴ τι δὲ ἀργύρια ἔχεις παρὰ σοὶ ἢ ὁλοκόττινα ἐν
if any.acc **prt** silver.coin.acc.pl have.2.sg with you.dat or gold.coin.acc.pl in
τάχει ἀπόστιλο(ν)·
speed.dat send.imp.aor

'**(And)** if you have any silver or gold coins with you, send them fast.' (the conjunctive particle δέ was annotated as dependent on the conditional clause, εἴ ἔχεις "If you have", while it was dependent on the main verb ἀπόστιλο(ν) "send" in the gold data)

(3) τοῦτο γὰρ ἐκέλευσεν ὁ κύριός μου Σύρος
this.neut.sg since command.aor.3.sg **the.nom master.nom** I.gen **Syros.nom**

'For my **master Syros** commanded this'. (Σύρος 'Syros" was annotated as an apposition of ὁ κύριός "the master", while it was the other way around in the gold data)

(4) Αὐρήλιος Ὀνήτωρ Αὐρηλίωι Φανίᾳ τῶι ἀξιολογωτάτωι
Aurelius.nom Onetor.nom Aurelius.dat Fanias.dat **the.dat** renowned.dat.sup
χαίρειν
be.happy.inf

'Aurelius Onetor greets **the** most renowned Aurelius Fanias'. (τῶι "the" was annotated as the head of ἀξιολογωτάτωι "the most renowned", while its head was Αὐρηλίωι Φανίᾳ "Aurelius Fanias" in the gold data)

(5) παρακαλῶ δὲ καὶ σπουδήν τινα πλείω προστεθῆναι Διοσκόρῳ
ask.1.sg prt and zeal.acc any.acc more.acc **bestow.inf.aor.pass** Dioskoros.dat
τῷ θαυμασίῳ, ὥστε κἀμὲ χρήσιμον αὐτῷ φανῆναι
the.dat marvelous.dat **so.that** and.I.acc useful.acc he.dat appear.inf.aor.pass
'And **I ask** you **to bestow** a little more zeal upon the marvelous Dioskoros, **so that** I too appear useful to him (. . .)' (ὥστε "so that" was annotated with παρακαλῶ "I ask" as its head, while its head was προστεθῆναι "bestow" in the test data)

Even though we have already undertaken some homogenisation efforts (see Section 4.1), the figures in Table 9 make clear that there is still quite some work left. Additional efforts and more detailed annotation guidelines (including better communication among the different treebank annotation projects) may further reduce such inconsistencies, even though we believe it will be impossible to eradicate all these issues: as linguistic categorisation is rather fluid, inconsistencies are often inherent to the annotation process. Moreover, it is important to underline that such consistency errors are only truly problematic for specific uses of the data. When the annotated trees are employed in a reading environment (see Section 6), consistency issues, such as in (2–5), do not pose real problems, given that humans will not stumble over divergent annotations (to the point of not even noticing them). When used as a corpus resource, inconsistent trees can significantly cloud the outcome of a corpus query. However, provided that researchers are well aware of the different annotation formats for a particular construction (e.g. apposition), they can perform several queries on the data to retrieve all relevant examples, thus factoring in the inconsistencies. On the other hand, when the syntactic trees are processed in a fully automatic way (e.g. to train new parser models, or to build distributional vectors on them, see Section 6), these inconsistencies become much more problematic, as they introduce a significant level of formal distinctions in the data not related to a real linguistic difference. Nevertheless, even there noisy trees may lead

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

to better results rather than not using any syntactic information at all, as shown by Keersmaekers (2020c).

Another category of problems are simply mistakes in the gold data, which were actually annotated correctly by the parser (10%). This indicates that the parser may also be used as part of a homogenisation action (which should be conducted with caution so as to prevent us from creating biased results; cf. in this respect also Dickinson 2015: 122). Next, there was a small group of problems that were impossible for the parser to solve (4%, or 20/500), as these were sentences in which multiple nodes were at the top of the tree in the gold data, while the output of the Stanford neural parser always had only one node at the top of the tree (see Section 5.1.8 for more detail). Although we used some rule-based techniques to ensure that most trees had only one head, due to the removal of ellipsis this could not completely be avoided (see Figure 1 in Section 4.2 for an example). These problems may therefore either be resolved by further reducing the number of elliptic structures in the data (see Section 5.1.8) or by using a parsing method which allows for multiple nodes at the top of the tree.

Some other problems were related to damaged text (3%, or 13/500), as described in Section 3.3. Although this category of problems is relatively small, the number of sentences that are damaged in the test set we analysed was rather small as well (16/1027). In the full dataset, the LAS of sentences that include damage (0.784, or 2,449/3,129) was substantially lower than for undamaged sentences (0.858, or 12,435/14,486 for undamaged sentences). This shows that more special care is needed to improve the analysis of such sentences: nevertheless, given the considerable problems involved with these damaged sentences, the LAS is perhaps higher than one would expect, suggesting that the method used here (i.e. simply treating them as normal sentences, see Section 4.2) has some merits.

Finally, there was a small set of errors related to true ambiguity, that is the sentence structure could be interpreted in at least two ways, and it was annotated in one way by the parser and in another way in the gold data. An example is given in (6) (which repeats [5]), in which the ambiguity is also present in the English translation: the sentence could either be interpreted as [appear [useful to him]], that is αὐτῷ 'he' depends on χρήσιμον 'useful', or as [appear [useful] [to him]], that is αὐτῷ 'he' depends on φανῆναι 'appear'. It goes without saying that in this case the correct interpretation is simply a matter of preference (in contrast with the 'consistency' issues, in which the same syntactic structure is annotated in different ways: in this truly ambiguous case two different syntactic structures/meanings simply share the same linguistic form).

(6) παρακαλῶ δὲ καὶ σπουδήν τινα πλείω προστεθῆναι Διοσκόρῳ
ask.1.sg prt and zeal.acc any.acc more.acc bestow.inf.aor.pass Dioskoros.dat
τῷ θαυμασίῳ, ὥστε κἀμὲ χρήσιμον αὐτῷ φανῆναι
the.dat marvelous.dat so.that and.I.acc useful.acc **he.dat** appear.inf.aor.pass
'And I ask you to bestow a little more zeal upon the marvelous Dioskoros, so that I too appear useful **to him** (. . .)'

When it comes to the 'real' syntactic mistakes, this set is also relatively diverse (we distinguished 48 different categories). Some of these errors, however, are more frequent than others. Starting with problems related to the syntactic **relation**, the most frequent interchange (28/277) was between adverbial (ADV) and complement (OBJ). This can also be seen in the confusion matrix in Figure 6: 9% of adverbials were annotated as complement (216/2408), and 3% of complements as adverbial (101/3155). In general the distinction between adverbial and complement is rather contentious in linguistics (the Universal Dependencies treebank, e.g.„ has not adopted it in its standard relations). Complements are generally considered to be required by the main verb, as shown in (7 and 8): the phrase εἰς ἅπαντας τοὺς συμπολίτας "for all our fellow citizens" in (7) is not required by ἔχεις "you have" (and accordingly it can be left out: "As we heard the goodwill that you have“), while the prepositional group εἰς Ἀντινόου “to Antinoopolis” in (8) is required by the verb ἀπῆλθεν “go away”. As prepositional groups with εἰς frequently express a direction, which is typically a complement, the parser made the wrong analysis of ‘OBJ’ rather than ‘ADV’ in sentence (7) (while it made the right choice in [8]). As the question of whether a constituent is obligatory or not depends on the meaning of the main verb, adding more or better semantic information to the parser (or even performing joint syntactic analysis and semantic role labelling) may therefore further improve parsing results. For human processing of the automated analysis, these types of errors are not particularly problematic, as the distinction complement/adverbial is often rather fluid to start with.

(7) ἡμεῖς ἀκούοντες τὴν εὔνοιαν ἣν εἰς ἅπαντας το[ὺς

we.nom hear.ptc.nom.pl the.acc goodwill.acc rel.acc **to all.acc.pl the.acc.pl**

συμπο]λίτας ἔχεις

fellow.citizen.acc.pl have.2.sg

‘As we heard the goodwill that you have **for all our fellow citizens** (. . .)’

(8) ἡ μήτηρ μου Θαῆσις εἰς Ἀντινόου, δοκῶ, ἐπὶ κηδίαν

the.nom mother.nom I.gen Thaesis **to Antinoos.gen** think.1.sg for funeral.acc

ἀπῆλθεν.

go.away.aor.3.sg

‘My mother Thaesis went **to Antinoopolis**, I think, for a funeral.’

As for the syntactic **head**, many problems are quite straightforward: they simply involve sentences where there are multiple theoretically possible candidates for a word’s head (e.g. verbs to which a noun can be attached) for which the parser made the wrong choice (66/277 grammatical mistakes are of this nature). This is illustrated in sentence (9), in hich the parser interpreted πόλλ’ “many, much” as an agreeing modifier with τὰ γράμματα “**many** letters”, rather than as an adverbial accusative with ὠφελήσει “help him **much**” (note that the plural noun γράμματα may either refer to one single letter or multiple letters, so both interpretations are theoretically possible). These errors are much more problematic than the ones mentioned above, as Plank *et al.* (2015) have also shown that humans attach more importance to the identification of the syntactic head rather than the relation. While in some cases, especially with complements, the semantics of the head may help to attach the word correctly, in many cases, such as in (9), only the wider discursive context or world knowledge may lead to the correct choice: as the rest of the text makes clear that the writer is referring to the letter that he is currently writing rather than several letters that he had written before, the interpretation ‘many letters’ in (9) would make little sense. Obviously without extensive pragmatic and encyclopaedic knowledge and with an analysis limited to the sentence rather than the text level such issues are impossible to solve.

(9) ἀλλὰ οἶδα ὅτι καὶ ταῦτά μου τὰ γράμματα πόλλ’

but know.1.sg that and this.nom.pl I.gen the.nom.pl letter.nom.pl **many.acc.pl**

αὐτὸν ὠφελήσει

he.acc.sg help.fut.3.sg

‘But I know that my letter will help him **much** (. . .)’

Another large number of problems is related to the morphological and part-of-speech analysis (34/277). These can be divided into two categories: a category in which the part-of-speech or morphological analysis was wrong and accordingly the parser was misled into a wrong analysis (19/34), and a category in which it was analysed correctly but this was ignored by the parser (15/34), for example when attaching an adjective to a non-agreeing word, as in (10), in which ϑειοτάτου 'most divine' was attached to βασιλείας 'reign' (even though it is masculine and βασιλείας feminine) rather than δεσπότου 'master'. As the parser does not use any rules to prevent modifying adjectives to be attached to a non-agreeing head, such combinations occasionally occur. Probably adding extra linguistic constraints to the parser (e.g. reject outputs with such non-agreement patterns) may further improve parsing results (see e.g. Ambati 2010). Alternatively, as the relation 'ATR' (attribute) is used for a wide range of agreeing and non-agreeing modifiers (e.g. relative clauses, genitive nouns, modification with a prepositional group, etc. as well), perhaps this makes it too difficult for the parser to 'learn' that adjectival attributes show a very strong tendency not to be combined with a non-agreeing head. Labelling agreement patterns in the training data (i.e. distinguishing between agreeing and non-agreeing modifiers) may therefore already solve this issue for the parser.

(10) βασιλείας τοῦ ϑειοτάτου καὶ εὐσεβεστάτου ἡμῶν δεσπότου
reign.gen the.gen **divine.gen.sup** and pious.gen.sup we.gen master.gen
Φλαουίου Ἰουστινιανοῦ
Flavius.gen Justinianus.gen
'During the reign of our **most divine** and pious master, Flavius Justinianus (. . .)'

The other grammatical errors are relatively diverse. Some common syntactic structures which gave way to a large number of errors involve appositive nouns (21/277 mistakes), coordination groups (11/277), subjects or objects of infinitives (7/277),[k] parenthetical (7/277) and relative clauses (6/277). In the future, we plan to undertake more concentrated efforts to tackle these grammatical problems (in the case of coordination, it was already addressed to a large extent, as shown in Section 5.1). In addition, there were also thirteen cases of a construction typical of the papyri, but not very common in literary Greek – these may only be resolved by expanding the papyrus training data. Finally, twenty-one errors turned out to be the result of other errors in the analysis, showing that a wrong analysis early on can create a snowball effect.

6. Putting the treebanks to work

The automated analysis of Greek texts creates a vast number of valorisation opportunities. To start with, we are already applying our NLP-pipeline to accelerate the expansion of the existing treebanks: it was the express goal of our ongoing project not only to achieve better parsing accuracy but also to offer tangible deliverables. This procedure has led to the creation of a wide range of new annotated Greek texts, developed in keeping with the guidelines of the AGDT and covering multiple genres and periods, including both literary texts and documentary papyri. By the end of 2020, the *Pedalion* corpus amounts to 300K tokens (see also Table 1). It proved much easier to manually correct a computer-generated annotation than to start an annotation entirely from scratch. The creation of new treebanks on the basis of pre-tagged and pre-parsed versions also allowed us to trace strengths and weaknesses of the parser as well as to make an analysis of inconsistencies. In the beginning of 2021, we launched *GLAUx* (the Greek word for 'owl', but also the acronym of "the **G**reek **L**anguage **AU**tomated"). *GLAUx* has released an extensive Open Access

[k] As the subject of an infinitive is expressed with the accusative in Greek, it is often difficult for the parser to predict whether such an accusative is a subject or an object. Possibly additional semantic information (i.e. certain nouns are more likely to be subjects rather than objects of certain verbs) may further improve the results.

corpus of automatically parsed Greek texts (Keersmaekers 2021).[l] This will enable colleagues to be much more efficient in building new treebanks. In parallel, the automatic annotations for the papyri are released through the Trismegistos project (Depauw and Gheldof 2013), which makes textual data and metadata for the papyri available for ancient historians and linguists. The online reading environment already includes morphology and lemmas, as well as historical data[m] – in the future the manually corrected syntactic information of the Pedalion treebanks will also be incorporated, making these understudied texts further accessible to a broader public.

Additionally, our Greek texts can be browsed tree by tree through the visualisation possibilities of the Perseids project.[n] In the future we plan to convert a number of our texts to fully-fledged automatically generated reading commentaries, highlighting important morphological and syntactic information for learners of Greek, which will be based on the new Perseus Scaife viewer. But it is important to emphasise that this massive number of automatically annotated texts can also be useful even before they are corrected manually. Three application possibilities stand out: linguistics, education and humanities research in general.

First, such a corpus enables **linguists** to perform detailed queries based on lemmatic, morphological and syntactic criteria. Obviously the results of the queries should be interpreted with due care, as errors cannot be excluded, but the quantity of the data involved (almost 40 million tokens) will clearly be a huge boon for research in Greek diachronic syntax. Second, we think it is feasible to transform this linguistic corpus into an **educational corpus**: a fully annotated corpus of Greek texts makes it easier for learners to delve deeper into any Greek text of their interest. A recent realisation is the development of Neorion, an introductory course to Ancient Greek for starters, which is based on authentic treebanked sentences.[o] The use of corpora for educational goals is overall well studied, with sufficient attention paid to opportunities, limitations and difficulties (see, e.g. Boulton 2009; McEnery and Xiao 2011), even though there is a clear bias towards English (Vyatkina and Boulton 2017: 5), while classical languages seem to have been excluded entirely from the discussion (see Van Hal and Keersmaekers 2022 for more details).

In addition, a syntactically annotated corpus can be highly relevant for **humanities research** in general, including fields such as history, philosophy and theology. The use of (corpus-)linguistic insights and methods in humanities research is currently on the rise. However, research mainly focuses on English texts (see, e.g. Zinn and McDonald 2018), and not at all on ancient Greek. Such research often concentrates on the use of n-grams (McMahon 2014: 26), and thus greatly hinges on (undeclined) words in a fixed sequence (Foxlee 2015). The GLAUx corpus approach allows researchers to exceed the mere word-based research by paying much more attention to word groups, syntagmata, collocations and constructions. Combined with the incorporation of distributional semantics into the corpus (Keersmaekers 2020c), which enables queries on the basis of word similarity and the automatic clustering of different meanings and usages of a lemma, we believe that GLAUx can contribute to solving urgent problems in present-day humanities research. One example is the domain of conceptual history, a research line that we plan to further explore in the future: recently, several scholars have stressed the importance of long-term studies on the history of concepts, avoiding a restrictive or 'pointillist' view of conceptual history (Armitage 2012: 498; McMahon 2014: 23–26). A corpus-based approach, exceeding the mere word level, can present a viable solution enabling historians to investigate conceptual trends and changes over time. Of course, it is in the interest of such kind of research that the annotated corpus contains as few errors as possible, but we think that significant results can already be obtained by relying on a corpus with the current parsing accuracy.

Finally, we have employed the automatic parses in several other natural language processing tasks: automatically generated syntactic information can significantly improve the quality of distributional word vectors (Keersmaekers 2020c) and is also beneficial for semantic role labelling

[l] See https://github.com/perseids-publications/glaux-trees for a demo version.
[m] See e.g. https://www.trismegistos.org/text/123.
[n] https://perseids-publications.github.io/pedalion-trees/
[o] http://www.pedalion.be/neorion

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

(Keersmaekers 2020a). In Keersmaekers (2020b) we also argued that syntactic information may improve the quality of morphological processing, as syntax and morphology are highly interrelated in Greek, even though we have not verified this hypothesis yet.

7. Conclusion and outlook

This paper has described a first attempt to automatically parse the documentary papyrus corpus, an extensive diachronic corpus of non-literary Greek (while also exploring possibilities to parse an even larger body of Greek text). We mainly carried out our experiments with the Stanford Graph-Based Neural Dependency Parser (Dozat *et al.* 2017), which could handle the complex morphology and free word order of Ancient Greek well. We have shown that through careful curation of the parsing data and several manipulation strategies, it is possible to achieve an LAS of about 0.85 for this corpus using the Stanford parser, even though the corpus shows some peculiar difficulties (e.g. damaged texts, a language, i.e. somewhat deviating from most of the training data) and even though other parsers might require other manipulations. In particular, we have shown that integrating a large training corpus of Greek text (involving some rule-based transformations) and homogenising its annotation, changing the annotation format for coordination structures, inserting morphological information in the parser model and (to a lesser extent) reducing the number of elliptic structures, along with some minor modifications, all lead to significant improvements in parsing accuracy. Even when the part-of-speech tags and morphology are automatically generated, there is only a low drop in parsing accuracy (LAS −0.006), provided that the training data uses automatically generated morphology as well. We have also shown that the current state-of-the-art model performs comparably on most literary texts, although some specific genres, that is scientific/philosophical and poetic texts, have lower parsing accuracy and require some special care in the future.

There is still room for further improvement, however. As discussed in Section 5.2, a large proportion of errors are related to inconsistencies in the training and test data. Advanced homogenisations, using a variety of techniques (e.g. rule-based homogenisation, anomaly detection; cf. also Dickinson 2015), are therefore necessary to improve the quality of the data. Nevertheless, such inconsistencies do not strongly prevent the data from being used in a wide array of applications, as shown in Section 6, as long as there is some level of human control involved. More crucial are the real grammatical parsing errors (which, as we have shown in Section 5.2, only constitute a little more than half of the total number of errors based on LAS). Several of these errors are caused by the linear nature of our parsing pipeline, in which morphological, syntactic and semantic analysis all build on each other (in this order), so that morphological errors may have repercussions for syntactic analysis, while the syntactic analysis cannot benefit from the results of the semantic analysis (e.g. semantic role labelling). The former problem is alleviated by Stanford Dependency Parser's use of character embeddings, which are able to capture the morphology of Greek to a great extent, even though not fully, as shown by the increase in accuracy when morphological information is added (see Section 5.1.9). On the other hand, as morphology and syntax are highly interrelated in inflectional languages such as Greek, the syntactic analysis may in turn substantially improve the quality of the morphological analysis (see also Keersmaekers 2020b).

More importantly, we have shown that a large category of parsing errors lie at the interface of syntax and semantics. The incorporation of higher quality word embeddings in the training model of the parser will therefore likely be able to substantially improve parsing results – and these embeddings, in turn, strongly benefit from syntactic information, see Keersmaekers (2020c), creating a self-reinforcing loop. Another possible way for further improvement is to perform syntactic analysis jointly with semantic role labelling, as several syntactic distinctions (e.g. adverbial vs. complement) are strongly intertwined with the semantic role of the given constituent (see Keersmaekers 2020a for the same problem from the perspective of semantic role labelling).

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Nevertheless, a significant number of parsing errors can only be resolved if the wider discursive context, or world knowledge is involved (e.g. by integrating information from knowledge databases, or by making a natural language processing system learn from real-world stimuli).

We have also shown that while most training data was literary, the parser was able to handle the papyri relatively well. Nevertheless, several constructions that were rather peculiar to these texts caused some problems. Ideally, we would be able to train a model by exclusively relying on papyrus texts (Mambrini and Passarotti 2012), but this is far from feasible from a quantitative perspective. One possible solution is to give more weight to those texts in the training corpus that have much in common with the new text that is to be analysed. This might, for example, be done by calculating text similarity on the basis of distributional models (see e.g. Turney and Pantel 2010).

It seems safe to state that the model can be further improved by adopting some additional strategies. In the future, we aim (1) to optimise the parser's parameters (we now made use of the default options), (2) to finetune the number of parts-of-speech distinguished (e.g. by also introducing quantifiers), (3) to design a special treatment for proper names (which are difficult to represent in word embeddings), (4) to better implement semantic information besides the word embeddings, (5) to reduce some 'noise' in the data by replacing non-standard or dialectical forms with their Classical Attic Greek counterparts and (6) to implement further error reduction strategies along the lines presented above. Our first experiments suggest that such measures have a favourable impact. We will also test more parsing systems in the future: in particular, we are currently experimenting with the transformer-based DiaParser. Transformer-based approaches have the advantage of no longer relying on pipelines, which are prone to error-propagation. Conversely, this also implies that the parser is no longer able to benefit from automated lemmatic and morphological annotations that reach relatively high accuracy.

The tools developed in the frame of our project are open source. Currently, we offer a first version of an extensive syntactically parsed corpus (GLAUx) via www.pedalion.be/trees, where our manually annotated trees can also be found. We are also collaborating with other scholars in the Perseus and Perseids projects to provide a version of our data in Universal Dependencies format. Within the framework of Trismegistos, we aim at developing a stable text infrastructure. The syntactic parsing model and the data used in this article can be reached through the GitHub account of the first author.

Financial support. This research was funded by Research Foundation Flanders (grants: 1162017N, G052021N).

Competing interests. The authors declare none.

References

Almas B. and Beaulieu M.-C. (2016). The perseids platform: scholarship for all. In Bodard G. and Romanello M. (eds), *Digital Classics Outside the Echo-Chamber: Teaching, Knowledge Exchange & Public Engagement*. London: Ubiquity Press, pp. 171–186.

Ambati B. R. (2010). Importance of linguistic constraints in statistical dependency parsing. In Proceedings of the ACL 2010 Student Research Workshop. Uppsala: Association for Computational Linguistics, pp. 103–108.

Armitage D. (2012). What's the big idea? Intellectual history and the Longue Durée. *History of European Ideas* **38**(4), 493–507.

Assael Y., Sommerschield T. and Prag J. (2019). Restoring ancient text using deep learning: a case study on Greek epigraphy. *ArXiv Preprint* ArXiv:1910.06262.

Ballesteros M. and Nivre J. (2012). MaltOptimizer: an optimization tool for MaltParser. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics*, pp. 58–62.

Bamman D. and Crane G. (2006). The design and use of a Latin dependency treebank. In *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT 2006)*, pp. 67–78.

Bamman D., Mambrini F. and Crane G. (2009). An ownership model of annotation: the Ancient Greek dependency treebank. In *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT 8)*, pp. 5–15.

Ben-David E., Cohen S., McDonald R., Plank B., Reichart R., Rotman G. and Ziser Y. (2021). Introduction. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pp. iii.

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Boulton A. (2009). Data-driven learning: reasonable fears and rational reassurance. *Indian Journal of Applied Linguistics* **35**(1), 1–28.

Buchholz S. and Marsi E. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*. Stroudsburg, PA: Association for Computational Linguistics, pp. 149–164.

Celano G. G. A. (2018). An automatic morphological annotation and lemmatization for the IDP papyri. In Reggiani N. (ed), *Digital Papyrology II*, pp. 139–147.

Celano G. G. A. (2019). The dependency treebanks for Ancient Greek and Latin. In Berti M. (ed), *Digital Classical Philology*. Berlin, Boston: De Gruyter, pp. 279–298.

Covington M. A. (1990). Parsing discontinuous constituents in dependency grammar. *Computational Linguistics* **16**, 234–236.

Depauw M. and Gheldof T. (2013). Trismegistos: an interdisciplinary platform for ancient world texts and related information. In *International Conference on Theory and Practice of Digital Libraries*. Cham: Springer, pp. 40–52.

Dickinson M. (2015). Detection of annotation errors in corpora. *Language and Linguistics Compass* **9**(3), 119–138.

Dik H. (1995). *Word Order in Ancient Greek: A Pragmatic Account of Word Order Variation in Herodotus*. Amsterdam: Gieben.

Dozat T., Qi P. and Manning C. D. (2017). Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL. 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Vancouver: Association for Computational Linguistics, pp. 20–30.

Elsahar H. and Gallé M. (2019). To annotate or not? Predicting performance drop under domain shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, pp. 2163–2173.

Feng S. Y., Gangal V., Wei J., Chandar S., Vosoughi S., Mitamura T. and Hovy E. (2021). A Survey of Data Augmentation Approaches for NLP. Preprint. Available at http://arxiv.org/abs/2105.03075

Fögen T. (2010). Female speech. In Bakker E. J. (ed), *A Companion to the Ancient Greek Language*. Chichester: Wiley-Blackwell, pp. 311–326.

Foxlee N. (2015). From analogue to digital: conventional and computational approaches to studying conceptual change. In *Conceptual Change: Digital Humanities Case Studies*, Helsinki.

Gorman V. B. (2020). Dependency treebanks of Ancient Greek prose. *Journal of Open Humanities Data* **6**(1), 1. https://openhumanitiesdata.metajnl.com/articles/10.5334/johd.13.

Hajič J., Bejček E., Bémová A., Buráňová E., Hajičová E., Havelka J., Homola P., Kárník J., Kettnerová V., Klyueva N., Kolářová V., Kučová L., Lopatková M., Mikulová M., Mírovský J., Nedoluzhko A., Pajas P., Panevová J., Poláková L., Rysová M., Sgall P., Spoustová J., Straňák P., Synková P. and Štěpánek M., Urešová J., Vidová Hladká Z., Zeman B., Zikánová D. and Žabokrtský Z. (2018). *Prague Dependency Treebank 3.5*. Institute of Formal and Applied Linguistics, LINDAT/CLARIN, Charles University, LINDAT/CLARIN PID. Available at http://hdl.handle.net/11234/1-2621

Harrington M. (2018). Perseids Project – Treebanked Commentaries at Tufts University. Available at https://perseids-project.github.io/harrington_trees/

Haug D. T. and Jøhndal M. (2008). Creating a parallel treebank of the old Indo-European bible translations. In *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, Marrakech, pp. 27–34.

Keersmaekers A. (2020a). Automatic semantic role labeling in Ancient Greek using distributional semantic modeling. In *1st Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2020)*, Marseille, 12 May 2020, pp. 59–67.

Keersmaekers A. (2020b). Creating a richly annotated corpus of papyrological Greek: the possibilities of natural language processing approaches to a highly inflected historical language. *Digital Scholarship in the Humanities* **35**(1), 67–82.

Keersmaekers A. (2020c). *A Computational Approach to the Greek Papyri: Developing a Corpus to Study Variation and Change in the Post-Classical Greek Complementation System*. PhD Dissertation, KU Leuven.

Keersmaekers A. (2021). The GLAUx corpus: methodological issues in designing a long-term, diverse, multi-layered corpus of Ancient Greek. In *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*. Association for Computation Linguistics, pp. 39–50.

Keersmaekers A., Mercelis W., Swaelens C. and Van Hal T. (2019). Creating, enriching and valorising treebanks of Ancient Greek: the ongoing Pedalion-project. In *SyntaxFest*, Paris.

Kübler S., McDonald R. and Nivre J. (2009). *Dependency Parsing*. San Rafael, CA: Morgan & Claypool Publishers.

Lee J., Naradowsky J. and Smith D. A. (2011). A discriminative model for joint morphological disambiguation and dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. **1**. Association for Computational Linguistics, pp. 885–894.

Luraghi S. (2003). Definite referential null objects in Ancient Greek. *Indogermanische Forschungen* **108**, 167–194.

Luraghi S. (2014). Typology of Greek. In Giannakis G. K., Bubenik V., Crespo E., Golston C., Lianeri A., Luraghi S. and Matthaios S. (eds), *Encyclopedia of Ancient Greek Language and Linguistics*, vol. **3**. Leiden: Brill, pp. 450–453.

Mambrini F. (2020). Daphne Trees. Available at https://perseids-publications.github.io/daphne-trees/ (accessed 2 April 2020).

https://doi.org/10.1017/S1351324923000384 Published online by Cambridge University Press

Mambrini F. and Passarotti C. (2012). Will a parser overtake achilles? First experiments on parsing the Ancient Greek dependency treebank. In *Eleventh International Workshop on Treebanks and Linguistic Theories*, EdiÇões Colibri, pp. 133–144.

McEnery T. and Xiao R. (2011). What corpora can offer in language teaching and learning. In Hinkel E. (ed), *Handbook of Research in Second Language Teaching and Learning*. New York: Routledge, pp. 364–380.

McMahon D. M. (2014). The return of the history of ideas? In McMahon D. M. and Moyn S. (eds), *Rethinking Modern European Intellectual History*. Oxford: Oxford University Press, pp. 13–31.

Mercelis W. (2019). *Syntactisch parsen van oudgriekse teksten. een vergelijkende studie.* Unpublished master thesis, KU Leuven - Faculteit Letteren, Leuven.

Nivre J., de Marneffe M.-C., Ginter F., Goldberg Y., Hajič J., Manning C., McDonald R., Petrov S., Pyssalo S., Silveira N. (2016). Universal dependencies, vol. 1: a multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, Portorož, pp. 1659–1666.

Nivre J., Hall J., Nilsson J., Chanev A., Eryiğit G., Kübler S., Marinov S. and Marsi E. (2007). MaltParser: a language-independent system for data-driven dependency parsing. *Natural Language Engineering* **13**(2), 95–135.

Nivre J. and Nilsson J. (2005). Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, MI: Association for Computational Linguistics, pp. 99–106.

Osborne T. (2019). *A Dependency Grammar of English: An Introduction and Beyond*. Amsterdam, Philadelphia: John Benjamins.

Plank B., Alonso H. M., Agić Ž., Merkler D. and Søgaard A. (2015). Do dependency parsing metrics correlate with human judgments? In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*. Beijing: Association for Computational Linguistics, pp. 315–320.

Popel M., Mareček D., Štěpánek J., Zeman D. and Žabokrtský Z. (2013). Coordination structures in dependency treebanks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers)*, pp. 517–527.

Porter S. E. and O'Donnell M. (2010). Building and examining linguistic phenomena in a corpus of representative papyri. In Evans T. V. and Obbink D. D. (eds), *The Language of the Papyri*. Oxford: Oxford University Press, pp. 287–311.

Robie J. (2017). Nine Kinds of Ancient Greek Treebanks. *Open Data for Digital Biblical Humanities*. Available at http://jonathanrobie.biblicalhumanities.org/blog/2017/12/20/treebanks-for-ancient-greek/

Schmid H. and Laws F. (2008). Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. **1**. Manchester: Association for Computational Linguistics, pp.777–784.

Seeker W., Farkas R., Bohnet B., Schmid H. and Kuhn J. (2012). Data-driven dependency parsing with empty heads. In *Proceedings of COLING 2012: Posters*. Mumbai: The COLING 2012 Organizing Committee, pp. 1081–1090.

Turney P. D. and Pantel P. (2010). From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research* **37**, 141–188.

Van Hal T. and Anné Y. (2017). Reconciling the dynamics of language with a grammar handbook. On Pedalion, an ongoing Greek grammar project. *Digital Scholarship in the Humanities* **32**(2), 448–454.

Van Hal T. and Keersmaekers A. (2022). Seeing the light through the trees: how treebanks can advance the education of classical languages. *Les Études Classiques* **89**, 349–372.

Verbrugghe G. P. (1999). Transliteration or transcription of Greek. *The Classical World* **92**(6), 499–511.

Vierros M. and Henriksson E. (2017). Preprocessing Greek papyri for linguistic annotation. *Journal of Data Mining and Digital Humanities*. Special Issue on Computer-Aided Processing of Intertextuality in Ancient Languages. Available at https://jdmdh.episciences.org/1385

Vilares M., Darriba V. M. and Vilares J. (2004). Parsing incomplete sentences revisited. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Seoul: Springer, pp. 102–111.

Vyatkina N. and Boulton A. (2017). Corpora in language teaching and learning. *Language Learning* **21**(3), 1–8.

Yordanova P. (2018). Treebank of Aphtonius' Progymnasmata. Available at https://github.com/polinayordanova/Treebank-of-Aphtonius-Progymnasmata

Zeman D., Hajič J., Popel M., Potthast M., Straka M., Ginter F., Nivre J. and Petrov S. (2018). CoNLL 2018 shared task: multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Brussels: Association for Computational Linguistics, pp. 1–21.

Zinn J. O. and McDonald D. (2018). Conceptual foundations. In Zinn J. O. and McDonald D. (eds), *Risk in The New York Times (1987–2014)*. Cham: Springer, pp. 7–65.

Cite this article: Keersmaekers A and Van Hal T. Creating a large-scale diachronic corpus resource: Automated parsing in the Greek papyri (and beyond). *Natural Language Engineering* https://doi.org/10.1017/S1351324923000384