Intelligenti Pauca
Probing a Novel Alternative to Universal Dependencies for Under-Resourced Languages on Latin

Daniel Couto-Vale
Kurfürstenstr. 148
10785 Berlin, Germany
danielvale@icloud.com

Konstantin Schulz
Humboldt University Berlin
Unter den Linden 6
10099 Berlin, Germany
konstantin.schulz@hu-berlin.de

Abstract

In this paper, we aim at improving the study of Latin in three ways: 1) by providing better visualizations of syntagma and structure for both research and the classroom, 2) by supporting a high-level search interface for corpus exploration, and 3) by improving the accuracy of taggers and parsers. To achieve this, we introduce a new linguistic description called Intelligenti Pauca, an alternative to Universal Dependencies for under-resourced languages. We show the key differences between the two linguistic descriptions, how the structure of Intelligenti Pauca favours our goals, and the effect it has on parsing accuracy for the Index Tomisticus Treebank.

1 Motivation

For Latin and Ancient Greek, researchers want to search for words and grammatical structures and view word features such as class and inflections (Monachini et al., 2018). Meanwhile, Latin and Greek teachers frequently make use of tools for visualizing and exploring grammatical structures in the classroom (Ellis, 2009; Mambrini, 2016; Augustinus et al., 2017; Guibon et al., 2020).

Annotated text corpora were built for implementing components for such tools including taggers, parsers, and searches (Abeillé, 2012, xiv), resulting in three dependency treebanks (Vincze et al., 2010, 1855): the Index Thomisticus Treebank (ITTB) (Passarotti, 2019), the Pragmatic Resources of Old Indo-European Languages (PROIEL) (Haug and Johndal, 2008), and the Ancient Greek and Latin Dependency Treebank (AGLDT) (Bamman and Crane, 2011). However, current tools for Latin present three issues: structures are 1. not highlighted, 2. unrelated to meaning (Khalili and Auer, 2013), and 3. often wrong (Monachini et al., 2018, 4), which is a problem for teaching (Müller and Oeste-Reiß, 2019, 59).

At the first frontier, attempts were made to represent features and grammatical structures graphically: e.g. adding information to a concordance line (Fischl and Scharl, 2014, 194) and showing a dependency tree for a sequence of words as in Figure 1, which reads «*However, women love chocolate desserts.*».

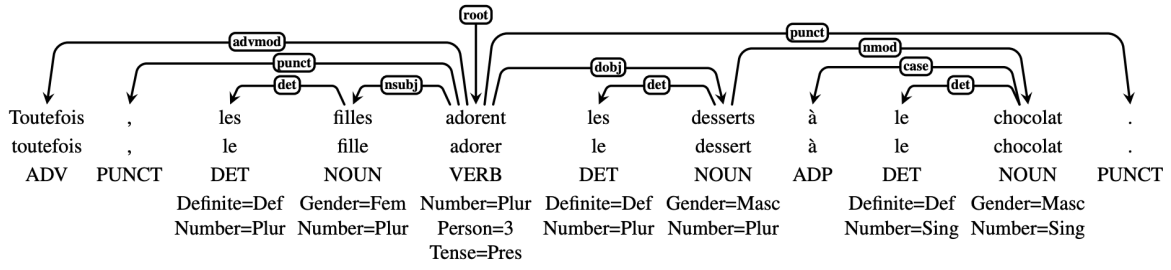

Figure 1: Visualization for Universal Dependencies (Nivre et al., 2016, 1660).

These visualizations are unsuitable for schools because they do not highlight grammatical structure in any way: e.g. frames or boundary markers. Highlighting is necessary to make structure observable (Arneson and Offerdahl, 2018, ar7,1), thus making it accessible to visual learners (Pitta-Pantazi et al., 2013, 201) and easier to process for all learner types (Kollöffel, 2012, 704). Besides, the grammatical structures here do not correspond enough to semantic structures for such a highlighting to be useful. In Section 2, we shall show how highlighting can be achieved and shallow semantic information displayed.

At the second frontier, Latin researchers need high-level searches for grammatical structures whereby they can answer their questions: a linguist may want to verify if 'medium-receptivity' ('middle' voice) as in *movetur* (*it moves*) and *movetur ab alio* (*it moves due to something else*) was the original meaning of *tur*-eding verbs; a theologian may be collecting evidence that a particular author assumed that three people emanate from God; a historian may be interested in how the actions of legates affected soldiers' morale; and a sociologist may want to know the relation between the origin of people's names and Roman identity. However, current search tools operate at a far too low level for them. In Section 4, we illustrate how to support high-level structural search for evidetiating such hypotheses.

Finally, we face an issue when improving parsing accuracy for historical languages: corpora will never increase. Here we must either improve generalization methods, annotations, or annotate more extant texts. In this paper, we focus on annotation improvement for better parsing accuracy.

Aiming at advances at these frontiers, we propose **Intelligenti Pauca** (IP), an alternative linguistic description to **Universal Dependencies** (UD, Nivre et al. (2016)), among others such as **Stanford Dependencies** (De Marneffe and Manning, 2008), based on a theory of ideational semantics by Halliday and Matthiessen (1999)[1]. Like UD, IP relies on dependency structures, not phrase structures, but it adds back a feature from the latter: the rank (Section 2.1), thus enabling visualization of grammatical structures (Goal 1) and corpus exploration (Goal 2). We converted UD annotations into IP. Since we needed to let dependency rules be learned from fewer examples (Goal 3), we aimed at reducing the number of rules that need to be learned for a particular labelled attachment (Goal 3.1) and reducing the number of features a rule is grounded on (Goal 3.2).

2 Intelligenti Pauca

Nivre's visualization does not highlight the grammatical structure and the amount of information it displays at once is far too great for the classroom. To improve it, we should aim at minimal intervention. The first step is to hide the structure and let only one layer of features visible without abbreviations as in Figure 2 (Goal 1). Structure and other features should be displayed only when needed[2][3]. The syntagma is highlighted by a frame, making it easier to understand for learners (Todi et al., 2018, 556).

cum	*ipse*	*deus*	*sit*	*nostrae*	*auctor*	*naturae*	.
conjunction	noun	noun	verb	noun	noun	noun	punctuation

Figure 2: Syntagma (ITTB, 198, 1)
«*since God himself is the creator of our nature.*»

However, this does not solve the whole issue. Grammatical structure must be highlighted and it must be meaningful. To achieve this, we can highlight the structure by framing it and reduce the number of dependencies shown at once, leaving only those that are related to each other semantically. In this way, we emphasize one aspect of meaning at a time, guiding viewers to comprehension (Goal 1). For instance, Figure 3 shows a structure with a lexical verb and two arguments. Here *Marker*, *Identified*, and *Identifier* are dependency labels and *Process* is the type of semantic element represented by the lexical verb.

Marker	Identified		Process	Identifier			Marker
cum	*ipse*	*deus*	*sit*	*nostrae*	*auctor*	*naturae*	.
conjunction	noun	noun	verb	noun	noun	noun	punctuation

Figure 3: Syntagma + Structure (ITTB, 198, 1)
«*since God himself is the creator of our nature.*»

[1]The available features and functions in the IP description are systematized in a SYS description, which can be imported as data into a database by a SYS description interpreter, also made available (JAR Scripts).

[2]Examples are referenced as (corpus, sentence id, word id).

[3]Some of the features such as 'seams' to be presented in Chapter 2.2 should be avoided in the classroom because they are meant to support the parsing mechanism and not to support teaching.

In this figure, we show only a selection of the dependencies and we provide labels from Halliday's theory of experiential semantics (Halliday and Matthiessen, 1999), which are more meaningful than the ones currently in use: namely, *Mark*, *Nsubj*, *Cop*, *Punct*. The resulting tabular visualization is easier to understand. Next, we explain how this visualization can be achieved with a dependency structure.

2.1 The rank

One way to reduce dependencies shown at once is to add **ranks** to dependency structures (Halliday, 1966). Ranks function as tags for grammatical units, indicating the type of phenomena units represent. There are three types of phenomena: figures are represented by **clauses**, sequences by **clause complexes**, and elements by **groups** and **phrases** (Halliday and Matthiessen, 1999, 48-49).

To add ranks to dependency structures, there must be an alignment between grammatical and semantic heads. In IP, auxiliary verbs such as *is* in *is coming* (*est* in *locutus est*) depend on the lexical verb 'in a verbal group' (Halliday and Matthiessen, 2014, 398) and other words depend on that verb 'in a clause' (Halliday and Matthiessen, 2014, 220). Participle verbs as in *the one moving* and *the one moved* (*illud movens* and *illud motum*) constitute a clause embedded 'in a nominal group' (Halliday and Matthiessen, 2014, 127). The same applies to other verbs linked to relative pronouns. Finally, lexical verbs depend on one another 'in a clause complex' (Halliday and Matthiessen, 2014, 428). This enables different visualizations: clause complexes as in Figure 4, clauses as in Figure 5, and groups as in Figure 6.

Extended					Extending					
veritatem	*meditabitur*	*guttur*	*meum*	,	*et*	*labia*	*mea*	*detestab...*	*impium*	.

Figure 4: Clause complex (ITTB, 2, 1)
«*My throat will judge the truth, and my lips will hate the wicked.*»

Phen.	Process	Senser		Marker
veritatem	*meditabitur*	*guttur*	*meum*	,

Marker	Senser		Process	Phen.	Marker
et	*labia*	*mea*	*detestab...*	*impium*	.

Figure 5: Clauses (ITTB: 2, 1; 2, 12)
«*My throat will judge the truth,*» «*and my lips will hate the wicked.*»

Thing	Possessor
guttur	*meum*

Thing	Possessor
labia	*mea*

Figure 6: Composed groups (ITTB: 2, 3; 2, 7)
«*my throat*» «*my lips*»

Ranked dependency structures differ from phrase structures because they can be discontinuous, thus there is no need to reconstruct word 'movements' (Mahajan, 2003, 218). However, both ranked dependents and phrases are semantic constituents, whereas non-ranked dependents are not. Given that discontinuities are frequent in Latin and Ancient Greek (Mambrini and Passarotti, 2012, 136), ranked dependency structures do not face the same challenges for these languages as phrase structure. Statistical dependency parsing with UDPipe (Straka et al., 2016) can produce ranked dependency structures as well as combinatory categorial parsing with OpenCCG (Bozsahin et al., 2005) and other parsing strategies.

2.2 Nouns, adjectives, numbers

Dependencies can be learned better if words share features in similar structures (Kübler and Hinrichs, 2001), especially **word classes** (Alfared and Béchet, 2012). Currently, dependencies in ITTB do not reflect meaning and word classes do not favour rule learning. IP solves these two issues by anchoring word classes onto types of represented elements (Goal 3.1). If the element being represented is a **simple thing** such as *guttur* (*the throat*) or *ego* (*me*), the word is a **noun**. Figures 7 and 8 illustrate the difference.

Head	Nmod
intelligere	*dei*
verb	propn
–	genitive

Head	Nmod
intelligere	*eius*
verb	pron
–	genitive

Amod	Head
suum	*intelligere*
adj	verb
nominative	–

Figure 7: UD – Modifiers (ITTB: 2049, 2; 2077, 12; 2050, 11)
«*God's intelligence*» «*his intelligence*» «*his intelligence*»

Thing	Possessor
intelligere	*dei*
noun	noun
–	genitive

Thing	Possessor
intelligere	*eius*
noun	noun
–	genitive

Possessor	Thing
suum	*intelligere*
noun	noun
genitive	–

Figure 8: IP – Possessors (ITTB: 2049, 2; 2077, 12; 2050, 11)
«*God's intelligence*» «*his intelligence*» «*his intelligence*»

In IP all **pronouns**, **proper nouns**, and **common nouns** are nouns because they represent things. Noun class is an extra feature. Pronouns such as *meum*, *tuum*, and *suum* (*my*, *your*, *his/her*) are pronouns, thus nouns, which differs from tradition (Oniga and Schifano, 2014, 95), and they are 'genitive' like other nouns with the same function (Rubenbauer and Heine, 2012, 54). They have a secondary case agreeing with the case of the modified noun, like adjectives do (Priscianus, 2010, 207). Agreement features are annotated as **seams** for both adjectives and such 'genitive' nouns inflected like adjectives. This lets the *Possessor* rule be heavily grounded on word class, subclass, and case (Goal 3.2).

In turn, **adjectives** represent additional **qualities** for simple things. Some function as classifiers in the nominal group (Halliday and Matthiessen, 2014, p.383), representing a more specific class of things than the noun represents on its own. This is the case of *pigmentaria* in *arte pigmentaria* (*the art of solution mixing*). Oftentimes, such a compound is synonymous to a noun such as *pigmentariae* (*the art of solution mixing*). UD treats those nouns as adjectives (see Figure 9), IP does not (see Figure 10). In turn, this separation between nouns and adjectives lets rules such as the *Classifier* rule be heavily grounded on word classes and subclasses (Goal 3.2).

Head	Amod
arte	*pigmentaria*
noun	adj

Head
pigmentariae
adj

Figure 9: UD – Modifiers & Heads (ITTB: 10, 4; 10, 26)
«*the art of solution mixing*» «*the art of solution mixing*»

Thing	Classifier
arte	*pigmentaria*
noun	adjective

Thing
pigmentariae
noun

Figure 10: IP – Classifiers & Things (ITTB: 10, 4; 10, 26)
«*the art of solution mixing*» «*the art of solution mixing*»

Thirdly, **numbers** such as *unus* (*one*), *primus* (*first*), *simplex* (*simple*), and so on represent a **quantity**. In UD, non-cardinal numbers are treated as adjectives. This poses an issue for the parsing of compound numbers such as *vigenti et unus* (*twenty one*), *vicesimus primus* (*twenty first*), *vigentuplex simplex* (*with twenty one parts*) and the like because rules cannot be learned across compounds in different number classes (see Figures 11).

Nummod	Amod	Amod	Head
unum	*simplex*	*suum*	*esse*
num	adj	adj	noun

Figure 11: UD – Modifiers (ITTB, 1482, 6)
«*his one simple being*»

Quantifer	Multiplier	Possessor	Thing
unum	*simplex*	*suum*	*esse*
number	number	noun	noun

Figure 12: IP – Quantifiers & Multipliers (ITTB, 1482, 6)
«*his one simple being*»

To solve this, in IP all numbers count as numbers as shown in 12. Numbers have different functions — e.g. Quantifier, Ordinator, Multiplier — depending on their class. Besides number class, numbers

also carry features for modulo and house: e.g. *unus* (*one*) is a 'cardinal' 'decimal' 'one' and *vicesimus* (*twentieth*) is an 'ordinal' 'decimal' 'ten'. These features enable compounding rules for different decimal houses within a numeric group and it enables different functions for different number classes (Goal 3.2). Figure 13 illustrates a compound quantity group in Latin.

Thousand				Hundred	Ten	Quantity
Hundred	Ten	Unit	House			
ducenta	*viginti*	*duo*	*milia*	*ducenti*	*viginti*	*unus*
number	number	number	number	number	number	number
hundreds	tens	units	thousands	hundreds	tens	units

Figure 13: IP – House, Hundred, Ten, Unit
«*two hundred twenty two thousand two hundred twenty one*»

In short, IP offers meaningful functions such as Classifier, Quantifier, Multiplier, and Possessor (also Ordinator, Deictic, Epithet) where UD offers only Modifier (Nmod, Amod, Nummod). In IP, word classes coincide with element types, which limits the number of rules (Goal 3.1), and they determine potential functions together with a small set of other features such as subclasses and cases (Goal 3.2).

2.3 Verbs

Transitivity A clause represents a figure composed of a process, participants, and circumstances (Halliday and Matthiessen, 1999, 128-172). The **lexical verb** represents the **process** in which things, qualities, and quantities take part. Let us consider the lexical verbs *habet* and *sit* in Figures 14 and 15.

Carrier	Marker	Process...	Attributor	...Process	Attribute	Marker
hoc	*autem*	*habet*	*aristoteles*	*pro*	*impossibili*	,
noun	adverb	verb	noun	adposition	adjective	punctuation

Figure 14: Transitive attributive clause (ITTB, 457, 1)
«*however, that was considered impossible by Aristotle,*»

Marker	Attribute	Process	Carrier		Marker
ut	*vehemens*	*sit*	*gaudium*	*eius*	.
conjunction	adjective	verb	noun	noun	punctuation

Figure 15: Intransitive attributive clause (ITTB, 154, 23)
«*that his joy is enourmous.*»

In these examples, *impossibili* (*impossible*) and *vehemens* (*enormous*) are attributes carried by, respectively, *hoc* (*that*) and *gaudium eius* (*his joy*). In turn, *Aristoteles* (*Aristotle*) is the person who attributes a quality to something. In IP, participant roles as in *Attribute*, *Carrier*, and *Attributor* are labelled instead of *Xcomp*, *Cop*, *Obj*, and *Nsubj*, thus enabling a visualization that guides readers towards a reasonable interpretation of transitivity (Goal 1) and high-level exploration of a corpus (Goal 2).

Verbal group Every time two or more verbs represent a single process, the lexical verb represents a process with participants and the others are **auxiliary verbs** (Halliday and Matthiessen, 2014, 396). Figure 16 contains such a verbal group with two verbs.

In Figure 16, the verb *est* (*must*) does not agree with the quantity of Actors nor with their role in speech. In addition, the Actor is represented by a genitive noun, not a nominative one typical of Actors (Menge et al., 2012, 383). This structure resembles that of more typical clauses with *ordinare* (*put order*), which shows that it is grounded more heavily on word classes such as nouns and lexical verbs than on inflectional features. On the one hand, the similarity in experiential semantics is an obvious improvement for visualization (Goal 1) and exploration (Goal 2). On the other, fewer rules (Goal 3.1) over fewer more general features (Goal 3.2) have a positive impact in parsing accuracy.

Marker	Actor	Process		Marker
		Auxiliary	Process	
quod	*sapientis*	*est*	*ordinare*	.
conjunction	noun	verb	verb	punctuation

Figure 16: Verbal group (ITTB, 5, 13)
«*because the wise must put order*»

Tense/mode Verbal groups can represent past, present, and future processes, the three **primary tenses** relative to 'now' (Halliday and Matthiessen, 1999, 214), in one of a few different **clause-linkage modes** (Whorf, 1956, 186). In free clauses, processes are placed in time in the injunctive mode as in the first column of Table 1. In bound clauses, clause-linkage modes realize types of logical relation together with conjunctions. Table 1 systematizes three modes of construing tense in Latin: here *ut* and *dum* are representatives of conjunctions used with conjunctive modes. Secondary tense (Halliday and Matthiessen, 1999, 399) such as 'past in past' in *moverat* (*had moved*) are left out for simplicity. Latin past verbs that oppose each other textually and interpersonally (Aerts, 2018) are placed in the same cell.

		conjunctive	
	injunctive	*ut*	*dum*
past	*movit, movebat, movet*	*moveret*	*movet*
present	*movet*	*moveat*	
future	*movebit*		

Table 1: Modes of construing primary tense in ITTB

Since these patterns are not covered by UD, current tools and components cannot determine primary tense. The root is also missed out because there are no features in UD for clause-linkage modes. In IP, this issue is solved by replacing traditional features by semantic and grammatical features, the latter being divided into group, word, and morpheme features. Table 2 shows morphemic features.

Verb	Aspect	Branch	Leaves	Verb	Aspect	Branch	Leaves
move ba t	*ō*	*bā*	*t*	*move ba t ur*	*ō*	*bā*	*tur*
move t o	*ō*	*ō*	*t*	*move t o r*	*ō*	*ō*	*tur*
move re	*ō*	*re*	–	*move ri*	*ō*	*rī*	–
mov it	*ī*	–	*it*	*mot um*	*ū*	–	*um*

Table 2: Stem aspect, branch, and leaves

There are three morpheme classes (Rubenbauer and Heine, 2012, 66-71): **Stem**, **Branch**, and **Leaf**. The available leaves depend on the selected branch, and the available branches depend on the selected stem aspect (Oniga and Schifano, 2014, 111). At the group rank, mode is partially determined by other words around the verbal group. For instance, if «*move t*» follows *dum*, it is *dum*-conjunctive, otherwise injunctive, a task modern taggers can do. Once a particular mode of construing primary tense is established, a primary tense can usually be determined solely based on the selection of verbs. This allows visualization of the tense (Goal 1) and searches for processes in particular primary tenses (Goal 2). Moreover, a parser can use the verbal modes in a verbal group together with conjunctions surrounding them to assess the chances that a particular lexical verb is the root of a dependency tree (Goal 3.2).

Finiteness In Latin, participants interacting in the dialogue such as *ego* (*I*) and *tu* (*you*) are usually left **implicit** if they are the subject (Oniga and Schifano, 2014, 209-213) (Rubenbauer and Heine, 2012, 115-116) and things that take part in two consecutive processes are left **elided** in the second clause (Kühner, 1879, 1042). **Finite** bound clauses are those that follow this pattern of implicitness and elision whereas **non-finite** bound clauses are those for which one participant is necessarily elliptic (Halliday and Matthiessen, 2014, 477). In Figure 17, we see three examples of non-finite bound clauses.

Marker	Phen.	Process
ad	*deum*	*cognoscendum*
adp.	noun	verb

(a) Non-finite verb seamed to *deum*

Marker	Phen.	Process
ad	*divina*	*cognoscenda*
adp.	noun	verb

(b) Non-finite verb seamed to *divina*

Marker	Process
ad	*ostendendum*
adp.	verb

(c) Unseamed non-finite verb

Figure 17: Non-finite bound clauses (ITTB: 121, 10; 238, 8; 563, 10)
«*to know God*» «*to know the divine*» «*to show*»

In UD, unseamed verbs such as *ostendendum* are 'gerunds' and seamed verbs such as *cognoscendum* are 'gerundives' and there is no feature that both have in common despite the fact that both gerunds and gerundives are *nd*-branch verbs. For every two rules that emerge from the examples in UD, IP lets one emerge by ascribing an *nd*-branch feature to these verbs (Goal 3.1). Departing from tradition (Rubenbauer and Heine, 2012, 202), it also makes the dependency between participants and processes be the same as in finite clauses, letting a single rule emerge from both finite and non-finite clauses.

Agreement The need for examples is further contained in IP by replacing original features (case, number, gender, person, tense, mode...) by word features for seam (agreement feature), and **foliage**, a set of leaves mapped to seams. Word-rank features result in matrices such as the one shown in Table 3.

	a-foliage	*am*-foliage	*ae*-*ī*-foliage	*ae*-*ō*-foliage	*ā*-foliage	
a-*am*-seam	*dic end a*	*dic end am*	*dic end ae*	*dic end ae*	*dic end a*	
um-*um*-seam	*dic end um*	*dic end um*	*dic end i*	*dic end o*	*dic end o*	unseamed
us-*um*-seam	*dic end us*	*dic end um*	*dic end i*	*dic end o*	*dic end o*	
ae-*ās*-seam	*dic end ae*	*dic end as*	*dic end arum*	*dic end is*	*dic end is*	
a-*a*-seam	*dic end a*	*dic end a*	*dic end orum*	*dic end is*	*dic end is*	
ī-*ōs*-seam	*dic end i*	*dic end os*	*dic end orum*	*dic end is*	*dic end is*	

Table 3: Gerunds and gerundives as *nd*-branch verbs

Gerunds and gerundives share the same stem aspect and an *nd*-branch (Rubenbauer and Heine, 2012, 71). In addition, all verbs following the adpositional marker *ad* in non-finite bound clauses have a leaf from the *am*-foliage, if they are seamed, or the *um*-leaf, otherwise.

Realization of conjunction	**Realization of seams**	
ad-conjunctive & seamed > *am*-foliage	*am*-foliage & *a*-*am*-seam > *am*-leaf	*am*-foliage & *ae*-*ās*-seam > *ās*-leaf
ad-conjunctive & unseamed > *um*-leaf	*am*-foliage & *um*-*um*-seam > *um*-leaf	*am*-foliage & *a*-*a*-seam > *a*-leaf
	am-foliage & *us*-*um*-seam > *um*-leaf	*am*-foliage & *ī*-*ōs*-seam > *ōs*-leaf

There is a total of 12 leaves for *nd*-branch verbs, five of which can occur in non-finite bound clauses with the adpositional marker *ad*. Twelve different verbs with two common feature (namely, aspect and branch) is a more general classification than 30 gerundives and 5 gerunds (Goal 3).

Potential seams can be determined based on morphemic features and contextual cues. The foliage can be determined based on the seam, if any, and contextual cues. Here, even if a word-rank tagging mistake is made at seam and foliage, the parser can still rely on the presence of an adposition such as *ad* and on lower-rank morphemic features such as *nd*-branch to determine that this is a non-finite clause. As a result, since the parser will count on fewer (Goal 3.2) more general (Goal 3.1) features, generalization will take place across examples with gerunds and gerundives for seldom adpositional markers.

Embedding Only some adnominal clauses in UD count as embedded clauses, namely those which contribute to reference. Embedded clauses are not logically related to other clauses directly, but rather modify a noun (Halliday and Matthiessen, 2014, 127, 382). In Latin, embedded clauses are either finite and have a 'relative' word[4] (Rubenbauer and Heine, 2012, 285, 287) or they are non-finite and have

[4] 'Relativsatz nach einer Einschränkung bzw. näheren Bestimmung bedürfenden Bezugswort'

a 'participle' verb[5] (Rubenbauer and Heine, 2012, 209-211). While in non-finite bound clauses verb foliage construes a type of logical relation together with adpositions, 'participle' verbs agree with the modified noun in case, thus they realize a case seam like adjectives do (see Figure 18).

Thing	Qualifier	
	Process	Goal
aliquod	*movens*	*se*
noun	verb	noun

(a) Operative embedded clause

Thing	Qualifier		
	Process		Actor
esse	*motum*	*ex*	*se*
noun	verb	adposition	noun

(b) Goal-receptive embedded clause

Figure 18: Embedded clauses (ITTB: 527, 10; 557, 16)
«*something moving itself*» «*a being moved by itself*»

Currently, the embeddedness of such clauses cannot be represented properly in UD. Nouns such as *aliquod* (*something*) are annotated as adjectival modifiers of verbs such as *movens* (*moving*), which are clausal subjects or objects of other verbs. In turn, nouns such as *esse* (*a being*) are annotated as auxiliaries of verbs such as *motum* (*moved*), which is a clausal subject or object of another verb. This categorial shifting generates instability between word classes and word functions. In IP the instability is reduced by having verbs in embedded clauses annotated as adposition-like modifiers of nouns (Goal 3.1). In this case, embedded clauses function as qualifiers within nominal groups as illustrated above.

Metaphor Finally, we come to the point where grammar 'folds on itself' (Halliday and Matthiessen, 1999, 227-293) (Halliday and Matthiessen, 2014, 659-707). We stop referring to *the thing moving* (*hoc movens*) or claiming that *this thing moves* (*hoc movetur*) and we start referring to *the mover* (*motor*) and *his motion* (*motus suum*). Examples of this can be found in Figure 19.

'Actor'	'Goal'
Thing	Possessor
motor	*universi*
noun	noun

(a) Actor as thing

'Process'	'Medium'	
Thing	Possessor	*metaphorical*
motus	*sui*	
noun	noun	

congruent

(b) Process as thing

Figure 19: Grammatical metaphor (ITTB: 19, 5; 381, 12)
«*the mover of everything*» «*his motion*»

Parsing results for *the mover of everything* and *his motion* in IP will represent a thing possessed by another (the 'metaphorical' structure). Such a parsing result cannot be understood as a direct representation of our experience. In the first example, *the mover* is 'possessed' by *everything else* only metaphorically. It actually moves everything else. In the second, *the motion* is a 'thing' and is 'possessed' by *something* only metaphorically. It is actually a process affecting that thing, the affected medium.

A full analysis must include the 'congruent' structure, which we could achieve by carrying out a second parse on nominal groups. This second-level parser should rely not only on grammatical features, but also on the semantic features of the represented elements, such as a further classification of things ('classified thing', 'actor as thing', 'process as thing', etc.) and their functions in the first-level structure (Halliday and Matthiessen, 1999, 278-296). This would guide the second-level parser towards an interpretation of the transitivity packed within such nouns. This second level of interpretation will not be integrated in the initial version of the IP description (1.0), but rather in a subsequent release cycle.

2.4 Cohesive ties

Some word links are not dependencies and are better understood as cohesive ties between constituents of different grammatical units. The clause complex in Figure 20 illustrates two types of cohesive ties.

[5] 'Attributives Partizip' in Hofmann et al.'s description.

Actor	Goal	Circum.	Process	Marker	Goal	Circum.	Process
qui	*res*	*directe*	*ordinant*	*et*	*eas*	*bene*	*gubernant*
noun	noun	adverb	verb	conjunction	noun	adverb	verb

Figure 20: Elision & anaphora (ITTB, 4, 17)
«who straighten things up and drive them well»

In the first clause, *qui* (*who*) and *res* (*things*) play the roles of, respectively, Actor and Goal of the action. In the second clause, the actor is elided to avoid repetition. This means that *qui* (*who*) in the first clause plays the role of ElidedActor of *gubernant* (*drive*) in the second. Moreover, *res* (*things*) in the first clause is the Same thing As *eas* (*them*) in the second. Both of these are cohesive ties in IP.

In OWL (Antoniou and van Harmelen, 2004), one can specify inference rules over cohesive ties such as the ones in Table 4 and let reasoners such as FaCT++ (Tsarkov and Horrocks, 2006) or HermiT (Glimm et al., 2014) follow the logical chain for «Actor», «Goal», and «Carrier».

Actor → «Actor»	Goal → «Goal»	Carrier → «Carrier»
ElidedActor → «Actor»	ElidedGoal → «Goal»	ElidedCarrier → «Carrier»
SameAs ∘ «Actor» → «Actor»	SameAs ∘ «Goal» → «Goal»	SameAs ∘ «Carrier» → «Carrier»

Table 4: Inference rules in Protégé SuperPropertyOf syntax

While *qui* (*who*) is the Actor of *ordinant* (*put order*) and the ElidedActor of *gubernant* (*drive*), it is the «Actor» of both. Thus if such inferred functions are stored in a DB, a researcher can search for all actions carried out by a given person, not only for those where the person is mentioned by name in the clause. In turn, this elevates the level at which one can query a corpus structurally (Goal 2).

3 Operations

3.1 Converting treebanks

We specified an SQL schema called **Dependency Base** (DB), which enables multiple analyses to be stored in parallel for the same text (DB Scheme). Since all three treebanks are available as CONLL-U files at LINDAT/CLARIAH-CZ (Universal Dependencies 2.6), we implemented a command line script for importing the text and its UD analysis from a CONLL-U file into a DB (JAR Scripts) and another for exporting an analysis as a CONLL-U file. In this setup, CONLL-U files work as an exchange format.

We specified a language called DUX for implementing conversion scripts for dependencies[6] and we implemented a DUX interpreter as a command line script, which converts a text analysis from a source linguistic description (e.g. UD description) into a target linguistic description (e.g. IP description). The DUX interpreter adds the resulting analysis into the DB as a stand-off annotation (Celano, 2019, 150). Finally, we implemented the conversion script from UD to IP in DUX, which can convert 93% of the ITTB in its current version.

To align grammatical and semantic heads, we needed to swap the direction of some dependencies and changed other structures entirely. Word features are determined by both form and context.

3.2 Creating a better parser

Since a different set of features and functions (dependency labels) exists for IP and UD and words depend on each other differently, we need to compare how easy it is for a parser to learn how to analyze text according to each description. For that purpose, we exported 398 lines of ITTB-train and 198 of ITTB-dev as CONLL-U files for UD and IP descriptions (Parallel Annotation). We compared the two file pairs for 'anchors', a tuple composed of tail class, head class, and function, which allows us to estimate how much evidence there is for each attachment/labelling rule and how many rules there are. For the same corpus segment, UD has roughly twice as many anchors as IP (108:59) and its anchor frequency

[6] `https://github.com/DanielCoutoVale/Dependencies/tree/master/ittb-ip` . It does not produce cohesive ties when converting a dependency treebank.

distribution has a longer tail (see Chart 1). Parsing shows a much better unlabelled attachment score (UAS) and a marginally better labelled one (LAS) (see Table 5).

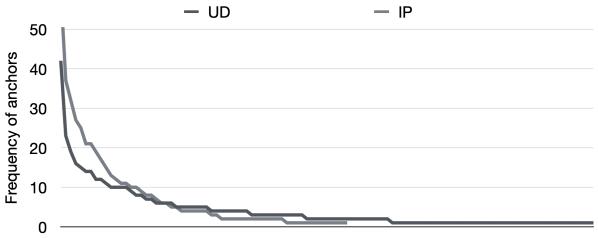

Chart 1: Anchor frequency distribution

	'Golden Tokens'		'Golden POS'	
	UAS	LAS	UAS	LAS
UD	28.40%	17.28%	36.42%	24.07%
IP	39.51%	19.75%	47.53%	26.54%

Table 5: Parsing scores

3.3 Creating a searchable resource

Treebanks are very expensive resources. One can progressively increase treebanks by adding verified parsing results to it, thus saving some time if one has a good parser. However, if this investment is not possible, one can automatically create an IP analysis layer with the parser above for the remaining texts, store the annotations in a DB, and search the DB for the desired syntagmata and structures. With that purpose in mind, we implemented a command line script for querying a DB (JAR Scripts). For generic search and visualization in IP resources, we plan to convert the provided CONLL-U files using Pepper (Zipser and Romary, 2010) and make them available in a public instance of ANNIS (Krause, 2019).

4 Exploring the resource

Once some IP annotations are stored in a DB, researchers can carry out high-level queries for a variety of research questions in different areas of humanities as illustrated in Table 6. Each square bracket stands for a word in the searched structure, the labels within it are word features, and the labels followed by parenthenses are links between words.

Linguistics	Theology
Did *or*-foliage 'passives' surpass *or*-foliage 'middles' in Latin? If so, when? (Kulikov and Lavidas, 2013)	How does Thomas Aquinas construe God as a single intelligence coming as three people? (Hillar, 2012)
[or-foliage goal-receptive verb] [or-foliage medium-receptive verb]	[number] [noun] Quantifier(1,2)[7] [number] [noun] Multiplier(1,2)
History	**Sociology**
Which actions carried out by the legates increased and decreased soldiers' morale? (Ureche, 2014)	How did people construe a Roman identity and Latin/Greek origins in Ancient Rome? (Elder, 2019)
[proper-noun] [adjective #legatus] Classifier(2,1)[8] [verb] [noun #Piso] «Actor»(2,1)[9]	[proper-noun] [verb] [noun #Corpus] «Carrier»(2,1)

Table 6: Research questions and corresponding corpus queries[10]

For UD-annotated corpora, there is no simple equivalent way to achieve this. For instance, there is no feature for the class of *or*-foliage verbs, no feature for non-cardinal numbers, no set of dependency labels and features associated with the roles of Actor and Carrier. For these questions, the regex-enabled search field found in web browsers might be a more suitable tool than a structural search in a UD treebank.

5 Conclusion

IP is a linguistic description based on Halliday's account of ideational semantics. In this paper, we showed that IP is more suitable than UD for three purposes: 1. visualizing syntagma and structure, 2.

[7] Views all numbers in context representing a quantity attributed to something.
[8] Collects all proper-nouns representing people classified as 'legatus'.
[9] Views all verbs in context representing actions carried out by Piso.
[10] *ElidedActor* and *SameAs* are IP ties, not dependencies. They are not included in the UD-IP conversion presented above.

enabling more detailed search in Latin corpora, and 3. annotating texts for creating taggers and parsers with UDPipe, while reducing the coarseness of functions in the representation. We also showed in which key ways IP differs from UD and explained how these differences improve the accuracy and utility of taggers and parsers in the study of Latin. The conversion script is available as DUX files (DUX Script) and a DUX interpreter is provided as a command line script (JAR Scripts).

Acknowledgements

This work is part of a project funded by the German Research Foundation (project number 316618374) and lead by Malte Dreyer, Stefan Kipf and Anke Lüdeling.

References

Anne Abeillé. 2012. *Treebanks: Building and Using Parsed Corpora*. Springer Science & Business Media, December.

Simon Aerts. 2018. Tense, aspect and Aktionsart in Classical Latin: Towards a new approach. *Symbolae Osloenses*, 92(1):107–149, January.

Ramadan Alfared and Denis Béchet. 2012. POS taggers and dependency parsing. *International Journal of Computational Linguistics and Applications*, 3(3).

Grigoris Antoniou and Frank van Harmelen. 2004. Web Ontology Language: OWL. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 67–92. Springer, Berlin, Heidelberg.

Jessie B Arneson and Erika G Offerdahl. 2018. Visual literacy in Bloom: Using Bloom's taxonomy to support visual learning skills. *CBE—Life Sciences Education*, 17(1):ar7,1–8.

Liesbeth Augustinus, Vincent Vandeghinste, Ineke Schuurman, and Frank Van Eynde. 2017. GrETEL: A tool for example-based treebank mining. In Jan Odijk and Arjan van Hessen, editors, *CLARIN in the Low Countries*, pages 269–280. Ubiquity Press.

David Bamman and Gregory Crane. 2011. The Ancient Greek and Latin Dependency Treebanks. In *Language Technology for Cultural Heritage*, pages 79–98. Springer.

Cem Bozsahin, Geert-Jan M Kruijff, and Michael White. 2005. Specifying grammars for OpenCCG: A rough guide. *Included in the OpenCCG distribution*.

Giuseppe GA Celano. 2019. An automatic morphological annotation and lemmatization for the IDP Papyri. In *Proceedings of the Third International Conference on Digital Access to Textual Cultural Heritage*, pages 149–153, Brussels, Belgium, May.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

Olivia Laura Elder. 2019. *Language and the Politics of Roman Identity*. Thesis, University of Cambridge, March.

Rod Ellis. 2009. Task-based language teaching: Sorting out the misunderstandings. *International journal of applied linguistics*, 19(3):221–246.

Daniel Fischl and Arno Scharl. 2014. Metadata enriched visualization of keywords in context. In *Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pages 193–196.

Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. 2014. HermiT: An OWL 2 Reasoner. *Journal of Automated Reasoning*, 53(3):245–269, October.

Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. 2020. When Collaborative Treebank Curation Meets Graph Grammars. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5291–5300.

Michael A.K. Halliday and Christian M.I.M. Matthiessen. 1999. *Construing Experience through Meaning: A Language-Based Approach to Cognition*. Continuum, London/New York.

Michael A.K. Halliday and Christian M.I.M. Matthiessen. 2014. *Halliday's Introduction to Functional Grammar*. Routledge, London/New York, fourth edition.

Michael A.K. Halliday. 1966. The concept of rank: A reply (1966). In *On Grammar*, pages 118–126. Continuum, London.

Dag TT Haug and Marius Johndal. 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)*, pages 27–34.

Marian Hillar. 2012. Thomas of Aquinas and the accepted concept of the trinity. In *From Logos to Trinity: The Evolution of Religious Beliefs from Pythagorians to Tertulian*, pages 249–272. Cambridge University Press.

Ali Khalili and Sören Auer. 2013. WYSIWYM Authoring of Structured Content Based on Schema.org. In Xuemin Lin, Yannis Manolopoulos, Divesh Srivastava, and Guangyan Huang, editors, *Web Information Systems Engineering – WISE 2013*, Lecture Notes in Computer Science, pages 425–438, Berlin, Heidelberg. Springer.

Bas Kollöffel. 2012. Exploring the relation between visualizer–verbalizer cognitive styles and performance with visual or verbal learning material. *Computers & Education*, 58(2):697–706.

Thomas Krause. 2019. ANNIS: A graph-based query system for deeply annotated text corpora. *Humboldt-Universität zu Berlin*, January.

Sandra Kübler and Erhard W. Hinrichs. 2001. From chunks to function-argument structure: A similarity-based approach. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*, pages 346–353, Toulouse, France. Association for Computational Linguistics.

Raphael Kühner. 1879. *Ausführliche Grammatik Der Latenischen Sprache*. Hansche Buchhandlung, Hannover.

Leonid Kulikov and Nikolaos Lavidas. 2013. Reconstructing passive and voice in Proto-Indo-European. *Journal of Historical Linguistics*, 3(1):98–121.

Anoop Mahajan. 2003. Word order and (remnant) VP movement. In Simin Karimi, editor, *Word Order and Scrambling*, pages 217–237. Wiley Online Library.

Francesco Mambrini and Marco Carlo Passarotti. 2012. Will a parser overtake Achilles? First experiments on parsing the ancient Greek dependency treebank. In *Eleventh International Workshop on Treebanks and Linguistic Theories*, pages 133–144. Edições Colibri.

Francesco Mambrini. 2016. The Ancient Greek Dependency Treebank: Linguistic Annotation in a Teaching Environment. In Gabriel Bodard and Matteo Romanello, editors, *Digital Classics Outside the Echo-Chamber: Teaching, Knowledge Exchange & Public Engagement*, pages 83–99. Ubiquity Press.

Hermann Menge, Thorsten Burkard, and Markus Schauer. 2012. *Lehrbuch der lateinischen Syntax und Semantik*. WBG, Wiss. Buchges, Darmstadt, 5., durchges. und verb. aufl edition.

Monica Monachini, Anika Nicolosi, and Alberto Stefanini. 2018. Digital Classics: A survey on the needs of Ancient Greek scholars in Italy. In *Proceedings of the CLARIN 2017 Conference*, pages 1–5. Linköping University Electronic Press.

Frederike Müller and Sarah Oeste-Reiß. 2019. Entwicklung eines Bewertungsinstruments zur Qualität von Lernmaterial am Beispiel des Erklärvideos. In Jan Marco Leimeister and Klaus David, editors, *Chancen und Herausforderungen des digitalen Lernens: Methoden und Werkzeuge für innovative Lehr-Lern-Konzepte*, Kompetenzmanagement in Organisationen, pages 51–73. Springer, Berlin, Heidelberg.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, and Natalia Silveira. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Renato Oniga and Norma Schifano. 2014. *Latin: A Linguistic Introduction*. Oxford University Press, Oxford.

Marco Passarotti. 2019. The Project of the Index Thomisticus Treebank. *Digital Classical Philology, De Gruyter, Berlin, Boston*, pages 299–320.

Demetra Pitta-Pantazi, Paraskevi Sophocleous, and Constantinos Christou. 2013. Spatial visualizers, object visualizers and verbalizers: Their mathematical creative abilities. *ZDM*, 45(2):199–213.

Priscianus. 2010. *Grammaire. [...] 1: Syntaxe Livre XVII*. Number 41 in Histoire des doctrines de l'Antiquité classique. Libraire Philosophique J. Vrin, Paris.

Hans Rubenbauer and Rolf Heine. 2012. *Lateinische Grammatik*. Buchner, Bamberg, [unveränderter nachdruck der] 12., korr. auflage 1995 edition.

Milan Straka, Jan Hajic, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *LREC*, pages 4290–4297.

Kashyap Todi, Jussi Jokinen, Kris Luyten, and Antti Oulasvirta. 2018. Familiarisation: Restructuring Layouts with Visual Learning Models. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 547–558, New York, NY, USA, March. Association for Computing Machinery.

Dmitry Tsarkov and Ian Horrocks. 2006. FaCT++ description logic reasoner: System description. In Ulrich Furbach and Natarajan Shankar, editors, *Automated Reasoning*, pages 292–297, Berlin, Heidelberg. Springer Berlin Heidelberg.

Petre Ureche. 2014. The soldiers' morale in the Roman army. *Journal of Ancient History and Archaeology*, 1(3), October.

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010. Hungarian dependency treebank. In *Seventh International Conference on Language Resources and Evaluation (LREC)*, pages 1855–1862.

Benjamin Lee Whorf. 1956. The relation of habitual thought and behavior to language (1939). In *Language, Thought and Reality: Selected Writings of Benjamin Lee Whorf*, pages 173–204. The MIT Press, Cambridge, MA.

Florian Zipser and Laurent Romary. 2010. A model oriented approach to the mapping of annotation formats using standards. In *Workshop on Language Resource and Language Technology Standards, LREC 2010*, May.