# Pre-annotation Matters: A Comparative Study on POS and Dependency Annotation for an Alsatian Dialect

**Delphine Bernhard, Nathanaël Beiner, Barbara Hoff**
Université de Strasbourg, LiLPa UR 1339
F-67000 Strasbourg, France
{dbernhard,n.beiner,barbara.hoff}@unistra.fr

## Abstract

The annotation of corpora for lower-resource languages can benefit from automatic pre-annotation to increase the throughput of the annotation process in a a context where human resources are scarce. However, this can be hindered by the lack of available pre-annotation tools. In this work, we compare three pre-annotation methods in zero-shot or near-zero-shot contexts for part-of-speech (POS) and dependency annotation of an Alsatian Alemannic dialect. Our study shows that good levels of annotation quality can be achieved, with human annotators adapting their correction effort to the perceived quality of the pre-annotation. The pre-annotation tools also vary in efficiency depending on the task, with better global results for a system trained on closely related languages and dialects.

## 1 Introduction

Automatic pre-annotation is often considered a cost-effective way of producing high-quality corpora, as it streamlines the process for human annotators. In the context of low-resource languages, pre-annotation can be a particularly beneficial practice, given that annotation tasks are often undertaken with limited human and financial resources. However, low-resource languages frequently lack training data or existing tools to obtain good quality pre-annotations.

In this article, we address the impact of pre-annotation on POS and dependency annotation in the Universal Dependencies (UD) framework (De Marneffe et al., 2021) for the Alsatian Alemannic dialects. Alsatian is a hypernym which refers to both Alemannic and Franconian dialectal varieties spoken in the Alsace region, in Northeastern France. The different Upper German dialects referred to by the term "Alemannic Alsatian dialects" are Northern Low Alemannic, spoken in the northern and central parts of Alsace, Southern Low Alemannic, spoken in the southern part of Alsace (south of Colmar), and High Alemannic, in the very south of the region. The Alemannic Alsatian dialects are closely related to other Alemannic German dialects, as for example Swiss German and Swabian, and to other dialectal varieties in the Oberdeutsch dialect family, as for example Bavarian.[1] Rhine Franconian is also spoken in the northwest of Alsace, but it is not included in our study, which focuses on Low Alemannic Alsatian. It is also worth mentioning that there is no consistent spelling standard for Alsatian dialects, which leads to high levels of variation in writing.

In this work, we compare three different pre-annotation methods, focusing on out-of-the-box tools that are easy to use without requiring extensive computational resources, advanced information technology skills or financial resources to pay for APIs. These methods rely either on tools trained for the closest standard language, German, or on a mix of German and related dialects, as well as an instruction-tuned generative large language model (LLM). Instruction-tuned LLMs have sparked the interest of researchers in recent years for annotation tasks with both positive and negative–or at least more cautious–conclusions. One of our goals was therefore to gain a better understanding of their advantages and pitfalls. We address the following research questions (RQ):

**RQ1** Is it possible to obtain good annotation quality with zero-shot pre-annotation only, when no existing tools are available for the target language?

**RQ2** Which pre-annotation method is the most useful?

**RQ3** Can pre-annotation bias be mitigated by using a mix of pre-annotation tools or, on the contrary, does it have a detrimental effect on quality?

[1] Alemannic Alsatian dialects appear under the name "Elsässisch", on the lower left of the map of German dialects by Werner König, published in the *dtv-Atlas Deutsche Sprache*, 17. edition, Munich 2011, p. 230-231.

*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 172–185
July 5–10, 2020. ©2020 Association for Computational Linguistics

**RQ4** What are the advantages and pitfalls of instruction-tuned LLMs for our target tasks?

## 2 Previous Work

### 2.1 Impact of Pre-Annotation

The impact of pre-annotation for treebank construction has been investigated since as early as 1993 (Marcus et al., 1993), with mostly consensual findings about the advantages of pre-annotation leading to a reduced annotation time, without negative effects on annotation quality. Some of the following papers nevertheless describe potential issues with automatic pre-annotations, in particular the influence of the pre-annotation tool on human annotators.

Fort and Sagot (2010) show that automatic pre-annotation for POS in English reduces the annotation time, even when pre-annotations have moderate levels of accuracy, and does not impact inter-annotator agreement or accuracy. But at the same time pre-annotation can introduce some systematic errors and biases, especially if the pre-annotation is rather good.

Berzak et al. (2016) describe the problem of *anchoring*, which they define as “a well known cognitive bias in human decision making, where judgments are drawn towards pre-existing values”, leading to a phenomenon that they call *“parser bias”*. They present a study to measure anchoring for POS tagging and dependency parsing in English and show that there is a bias towards the outputs of the specific pre-annotation tool being edited by the human annotators.

For languages other than English, Mikulová et al. (2022) investigate pre-annotation bias for Czech dependency syntax. They observe that annotations are more consistent when the data is pre-annotated, which might point at an influence of the automatic pre-annotation on the annotators. Overall, annotation is sped-up when the texts are pre-annotated and inter-annotator agreement improves.

The efficacy of automatic pre-annotation has also been studied in the context of languages characterised by a high level of variation in writing. Eckhoff and Berdičevskis (2016) train a parser for Old East Slavic and use it for pre-annotation in an experiment involving four annotators. Pre-annotation led to gains in speed, without apparently lowering annotation quality.

### 2.2 Zero-Shot Transfer of Taggers and Parsers across Languages and Varieties

Zero-shot[2] transfer has been proposed in recent years as a viable option for low-resource languages with neither existing taggers or parsers, nor big enough training corpora.

For POS tagging and dependency parsing, Lauscher et al. (2020) demonstrated that transfer performance is mainly influenced by the similarity in syntactic properties between the source and target languages. This finding was substantiated by de Vries et al. (2022), who explored zero-shot cross-lingual transfer learning using multilingual pre-trained models for POS tagging, with 65 source languages for training and 105 target languages for testing. They highlighted that including the target language, and to a lesser extent the source language, in the training dataset for the multilingual pre-trained model is particularly crucial. Vandenbulcke et al. (2024) confirmed previous observations that training on closely related languages is key. Transfer of parsers across different historical states of a language is investigated by Lücking et al. (2024), who show that parsers trained on contemporary English and German can be transferred to older language states with very modest drops in performance.

Methods have been proposed to improve tagger and parser efficiency in zero-shot transfer. To mitigate noise caused by spelling variations between source (training) and target (automatic annotation) languages, data transformation can be employed. Various automated methods have been suggested, typically utilizing data transformation techniques leading to an increased resemblance between source and target language data: phonemic and graphemic transformation rules (Hana et al., 2011), lexicon-based translation of words (Bernhard and Ligozat, 2013; Wang et al., 2022), random noise injection in training data (Aepli and Sennrich, 2022; Blaschke et al., 2023).

Finally, more recent work by Ezquerro et al. (2025) has investigated the use of generative large language models for zero-shot dependency parsing. They compared syntactic trees obtained via simple prompting of instructed-tuned LLMs against ran-

[2] Here we use the term zero-shot in the context of cross-lingual tasks, where a multilingual pre-trained model is fine-tuned on a language for a task and then directly applied on another language. Zero-shot is used in this sense by e.g. Aepli and Sennrich (2022), de Vries et al. (2022) and Vandenbulcke et al. (2024).

dom trees generated via different baselines. They reach negative conclusions, since most of the tested LLMs are not able to beat the strongest baselines.

## 3 Experimental Setup

### 3.1 Corpus

Our corpus consists of texts translated from French into Low Alemannic Alsatian and belonging to different genres and domains (see Table 1). Most of the sources were translated in the realm of our project, either by a professional translator or by a project participant. In addition, we included three sources with pre-existing translations into Low Alemannic Alsatian: the *Universal Declaration of Human Rights*,[3] which is already present in other Universal Dependencies treebanks, such as French ParTUT, the *Parable of the Prodigal Son* (Steiner and Matzen, 2016) and the *North Wind and the Sun* (Boula de Mareüil et al., 2018).

The corpus was tokenised using an adapted version of the tokenisation script developed by Blaschke et al. (2023) for Bavarian and split into 6 annotation batches. Each batch contains a number of sentences for each source that is proportional to the length of the corresponding source. The original sentence order is kept.

For the analysis presented here, we only retained sentences whose tokenisation was not corrected or modified during the manual annotation correction process, which would prevent the calculation of agreement scores with the pre-annotation. The tokenisation had to be corrected for e.g. contracted forms or epenthetic consonants. Table 2 details the number of sentences and words in each batch, for the analysed subset and in total.

### 3.2 Pre-annotation Methods

We compare three main pre-annotation methods, based on the analysis of zero-shot transfer methods in Section 2.2.

**UDPipe** (Straka, 2018) We used UDPipe 2 through the LINDAT UDPipe REST Service[4] and applied the two available German models: GSD (McDonald et al., 2013) and HDT (Borges Völker et al., 2019). Prior to annotating our corpus, we normalize accented vowels to their unaccented form and use a bilingual Alsatian-German lexicon of closed class words to translate Alsatian forms to their German equivalent (Bernhard, 2023). The aim of this pre-processing of Alsatian data is to make Alsatian look more like German and thus be able to use models trained on German directly, without retraining. We used the latest models available when performing the pre-annotation: for batches 1 to 4, the models trained on UD 2.12[5] were used, and for batches 5 and 6, the models trained on UD 2.15.[6] Only very slight changes in performance were reported between the two versions of the training data in the detailed model performance.

**Mistral Large** We used the free Mistral API with a prompt (see Appendix A) and two different temperature values: 0.1 and 0.7. The sentences were provided in the CoNLL-U format, with the requested annotations left empty. The Mistral Large model claims to excel in several languages, including German.[7] The prompt was refined during the course of the manual annotation period to correct minor details (typos, addition of relation subtypes based on evolutions of the guidelines). In addition to POS and dependency relations, the prompt also requested for a gloss in French. Since Mistral does not always output a correct CoNLL-U file, we semi-automatically corrected the following errors: extraneous POS and dependency annotations on multiword tokens, missing tokens and text metadata, spaces instead of tabulations, missing empty '_' columns. Moreover, the annotation sometimes fails unexpectedly for some sentences and the annotation was then retried. For one of the batches, we also experimented with "agents",[8] in order to decompose the annotation process in the following annotation steps: POS, French gloss and dependency relations, followed by a CoNLL-U format verification agent. The output of each agent was passed as input to the next agent.

**ArboratorGrew** trainable parsing service[9] on the ArboratorGrew annotation platform (Guibon et al., 2020). The parser (Guiller, 2020; Peng et al., 2022) is based on the architecture of Dozat and Manning (2017) and was trained using the test splits for the following UD corpora: 977 sentences from

[3] https://www.ohchr.org/en/human-rights/universal-declaration/translations/elsassisch?LangID=gsw

[4] https://lindat.mff.cuni.cz/services/udpipe/

[5] https://ufal.mff.cuni.cz/udpipe/2/models#universal_dependencies_212_models

[6] https://ufal.mff.cuni.cz/udpipe/2/models#universal_dependencies_215_models

[7] https://mistral.ai/fr/news/mistral-large-2407

[8] https://docs.mistral.ai/capabilities/agents/

[9] https://arborator.github.io/arborator-documentation/#/parser

| Title | Author | Domain | Genre | Sentences | Words |
|---|---|---|---|---|---|
| *Monday Tales* | Alphonse Daudet | Literary | Short story | 179 | 3,924 |
| *Universal Declaration of Human Rights* | United Nations | Legal | Official charter | 83 | 2,231 |
| *Decameron* | Boccace | Literary | Short story | 19 | 494 |
| *Peter and the Wolf* | Sergueï Prokofiev | Literary | Symphonic tale | 65 | 940 |
| *Parable of the Prodigal Son* | Luke | Religion | Parable | 29 | 631 |
| *The North Wind and the Sun* | Esope | Literary | Fable | 6 | 127 |
| *Chronicles on French Regional Languages* | Michel Feltin-Palas | Journalism | Column | 177 | 4,354 |
| | | | **TOTAL** | 558 | 12,701 |

Table 1: Corpus contents. “Words” refers to syntactic words.

| | Analysed part | | Total | |
|---|---|---|---|---|
| **Batch** | **Sent.** | **Words** | **Sent.** | **Words** |
| 1 | 74 | 1,670 | 88 | 1,978 |
| 2 | 88 | 1,771 | 93 | 1,967 |
| 3 | 84 | 1,672 | 92 | 1,972 |
| 4 | 85 | 1,769 | 93 | 1,957 |
| 5 | 89 | 2,248 | 94 | 2,380 |
| 6 | 91 | 2,220 | 98 | 2,447 |
| Total | 511 | 11,530 | 558 | 12,701 |

Table 2: Corpus batches.

| **Pre-annotation** | **Batch** | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| UDPipe-GSD | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| UDPipe-HDT | ✓ | ✓ | ✓ | – | ✓ | ✓ |
| Mistral temp=0.7 | ✓ | ✓ | ✓ | – | – | – |
| Mistral temp=0.1 | – | ✓ | ✓ | – | ✓ | ✓ |
| Mistral agents | – | – | – | ✓ | – | – |
| ArboratorGrew | – | – | – | ✓ | ✓ | ✓ |

Table 3: Distribution of pre-annotation settings across batches.

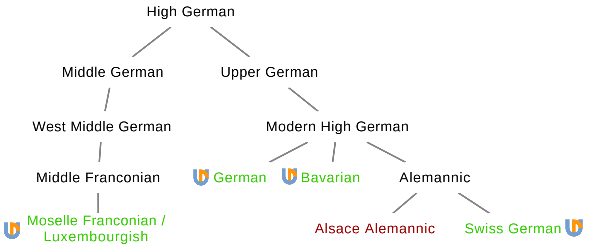


Figure 1: Simplified family tree of Alsace Alemannic based on Glottolog (Hammarström et al., 2024), with related languages available in UD.

German GSD (McDonald et al., 2013), 1,070 sentences from Bavarian MaiBaam (Blaschke et al., 2024), 100 sentences from Swiss German UZH (Aepli, 2018) and 20 sentences from Luxembourgish LuxBank (Plum et al., 2024). These languages were selected based on their proximity to Alsace Alemannic (see Figure 1). In addition, we added 25 Alsatian sentences which were annotated as examples for earlier versions of the annotation guide. In total, 2,192 sentences from 5 Germanic Languages were used to train ArboratorGrew. The Labelled Attachment Score (LAS) obtained during training was 0.83 (Epoch 55). Due to an unavailability of the parsing service during the first half of our annotation period, we started using ArboratorGrew only from batch 4 onwards.

**Selection of the pre-annotation** We randomly choose one of the available pre-annotations for each sentence and assign different pre-annotations to each annotator. This approach ensures that human annotators start from different pre-annotations, preventing any potential uniform and unique influence on their annotations. For each annotation batch, at least 3 different pre-annotation methods were used (see Table 3).

### 3.3 Manual Correction Process

The corpus was annotated by two annotators who are co-authors of this paper: A1 and A2. Both are native speakers of Alsace Low Alemannic, have obtained a master’s degree in linguistics and written Master theses on the Alsatian dialects. The initial guidelines had been drafted by one of the two annotators based on a study of existing grammars in Alsatian and existing POS annotation guidelines (Bernhard et al., 2018). Both annotators were given an initial training batch, which was used to make them familiar with the annotation tool and the guidelines. After each batch, the annotators discussed their annotations in order to reach a consensual `validated` annotation (see Figure 2 for an example validated annotation). The decisions reached during their discussions were also inte-

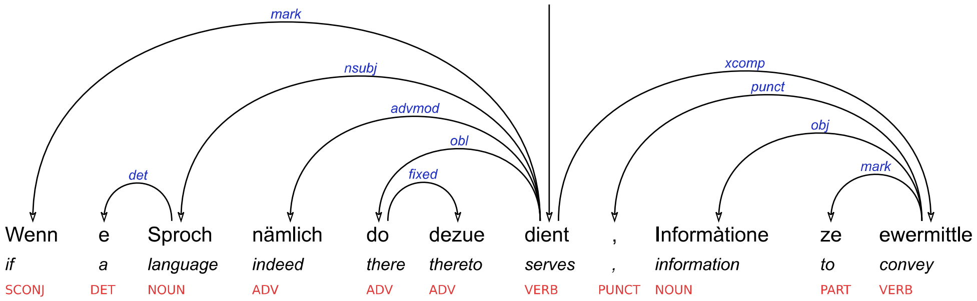


Figure 2: Example annotated sentence with English glosses.

grated in the annotation guide.[10]

The annotation tool was ArboratorGrew (Guibon et al., 2020): the pre-annotated CoNLL-U files were uploaded on the platform and then annotated in blind annotation mode. The whole annotation process reported in this paper took place over a period of four months.

### 3.4 Agreement Assessment

We used the following scores to measure agreement between pre-annotations, manual corrections and the final validated annotations:

**POS:** Cohen's $\kappa$ (Cohen, 1960) for POS labels, as well as accuracy.

**Dependencies:** Adaptation of Krippendorff's $\alpha$ (Krippendorff, 1970) to dependency relations proposed by Skjærholt (2014), as well as UAS (Unlabelled Attachment Score), LAS (Labelled Attachment Score) and LAcc (dependency Label Accuracy) (Eisner, 1996; Nivre et al., 2004; Buchholz and Marsi, 2006).[11]

## 4 Results

### 4.1 Results per Annotation Batch

Figure 3 shows the evolution of inter-annotator agreement over time for both tasks. The agreements tend to increase, with a steeper rise and a higher variability in agreement for dependencies. Agreement levels for POS are more consistent, indicating that the task is less difficult. Overall, the increase in agreement suggests that annotators improve their consistency over time, possibly due to improved guidelines, better training, or increased familiarity with the annotation task.

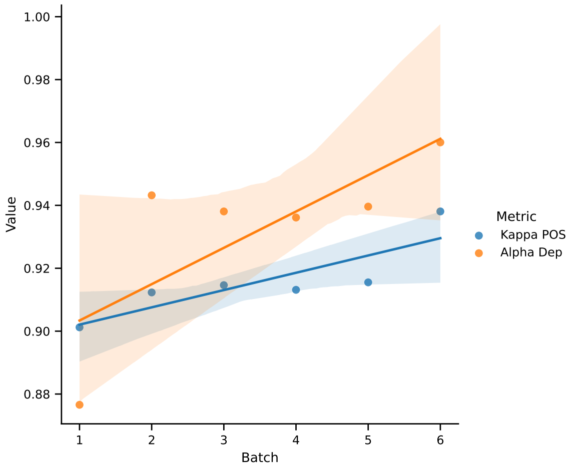


Figure 3: Evolution of inter-annotator agreement scores.

Figure 4 illustrates the evolution of the agreement between the annotators and the automatic pre-annotation over time. For A1, agreement with the pre-annotations remains relatively stable, with a slight downward trend. For A2, the declining trend is more marked for dependencies, with high variability, while for POS the agreement slightly improves. The decline in the agreement for dependencies is likely due to the quality of the automatic pre-annotations: over time, the annotators are more actively correcting errors. The difference in POS agreement trends between A1 and A2 could suggest varying levels of reliance on pre-annotations. Overall, both annotators align more with POS pre-annotations, while increasingly correcting errors in pre-annotations for dependencies.

Finally, Figure 5 illustrates the evolution of agreement between the two annotators and the validated annotation. Both annotators show an increasing agreement trend over batches, indicating an improvement in their annotation consistency over time. In contrast to Figure 4, agreement is consistently higher for dependencies than for POS: this might point at an over-reliance on POS pre-

[10] Details about the annotation guide and specific linguistic properties of the dataset will be described in another article.

[11] For all dependency measures, we reuse the scripts developed by Skjærholt (2014) and available at `https://github.com/arnsholt/syn-agreement/`. Similarly to (Dipper et al., 2024), we converted them to Python 3.

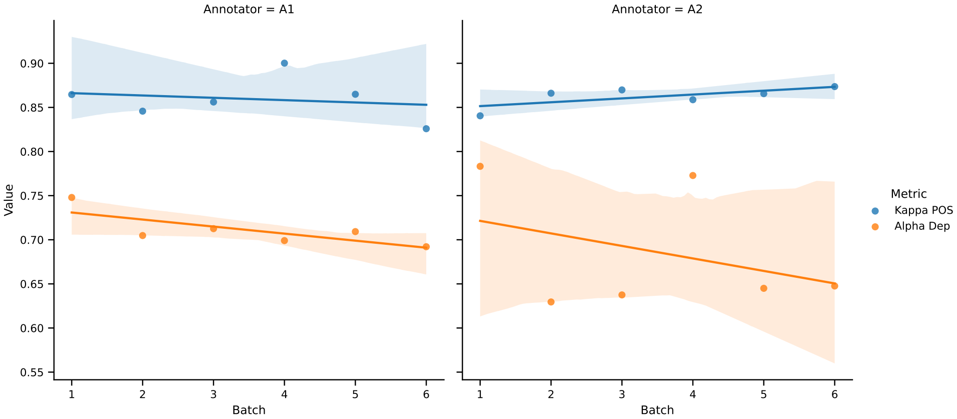


Figure 4: Evolution of agreement scores with respect to the pre-annotation.

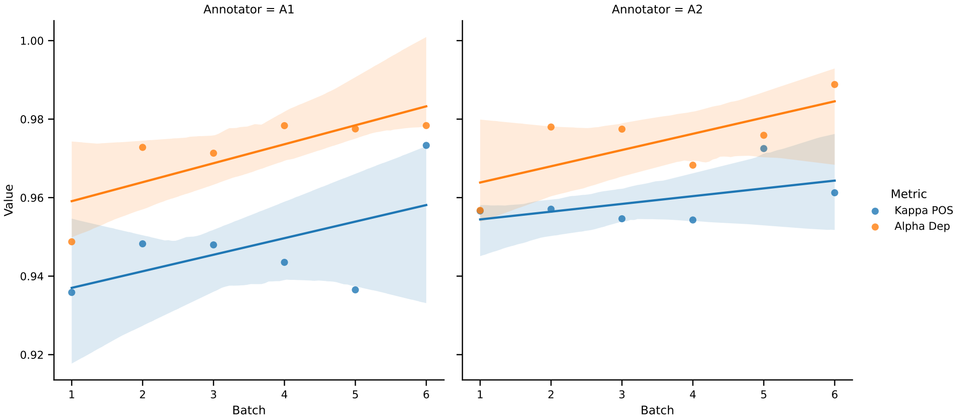


Figure 5: Evolution of agreement scores with respect to the validated annotation.

annotations, being perceived as good enough, and an under-reliance on dependency pre-annotations, being perceived as error-prone and deserving more corrections.

To conclude, lower agreements are observed with the pre-annotation and higher agreements with the validated version, with inter-annotator agreements in-between. This is a result of consensus building by the two annotators to reach the validated annotation (see Table 6 in Appendix D for the detailed agreement scores for each batch.). Overall, the inter-annotator agreements are high (POS $\kappa \geq 0.90$, dependency $\alpha \geq 0.88$), as well as agreements with the validated annotation (POS $\kappa \geq 0.94$, dependency $\alpha \geq 0.95$). Regarding **RQ1** (*Is it possible to obtain good annotation quality with zero-shot pre-annotation only, when no existing tools are available for the target language?*), our findings demonstrate that good levels of annotation quality can be attained even in the absence of pre-existing annotation tools for our target language. This suggests that relying on closely-related languages or multilingual LLMs can be a viable option in such cases. However, as we did not include a control setting in which the annotators started from scratch, we cannot compare the quality of the annotations with and without pre-annotation.

## 4.2 Analysis of the Pre-annotation Methods

Table 4 details the agreement scores broken down by pre-annotation method and Figure 6 displays the per-sentence POS accuracy and LAS with respect to the validated annotation for UDPipe-GSD, Mistral and ArboratorGrew. Mistral obtains the best results overall for POS, followed closely by ArboratorGrew. Both UDPipe models have lower levels of performance for this task. UDPipe-GSD obtains the best results for dependencies, both in terms of dependency attachments and dependency labels. ArboratorGrew also has good performance for this task, while Mistral obtains the lowest UAS and LAS. Interestingly, Mistral still gets good dependency label accuracy scores. Finally, the density plots in Figure 6 confirm that Mistral has a

| Pre-annotation | Annot. | Sent. | Tok. | K POS | Acc POS | $\alpha$ Dep | UAS | LAS | LAcc |
|---|---|---|---|---|---|---|---|---|---|
| UDPipe-GSD | A1 | 148 | 3,293 | 0.84 | 0.86 | 0.82 | 0.76 | 0.63 | 0.74 |
| | A2 | 125 | 2,815 | 0.82 | 0.84 | 0.83 | 0.79 | 0.63 | 0.71 |
| | validated | 273 | 6,108 | 0.80 | 0.82 | 0.79 | 0.76 | 0.60 | 0.70 |
| UDPipe-HDT | A1 | 144 | 3,218 | 0.79 | 0.81 | 0.73 | 0.64 | 0.53 | 0.64 |
| | A2 | 72 | 1,792 | 0.78 | 0.80 | 0.77 | 0.66 | 0.56 | 0.67 |
| | validated | 216 | 5,010 | 0.75 | 0.77 | 0.72 | 0.64 | 0.51 | 0.63 |
| Mistral | A1 | 149 | 3,126 | 0.93 | 0.93 | 0.62 | 0.60 | 0.52 | 0.73 |
| | A2 | 214 | 4,446 | 0.91 | 0.92 | 0.50 | 0.56 | 0.48 | 0.72 |
| | validated | 363 | 7,572 | 0.88 | 0.89 | 0.52 | 0.55 | 0.45 | 0.69 |
| ArboratorGrew | A1 | 70 | 1,713 | 0.89 | 0.90 | 0.64 | 0.74 | 0.62 | 0.74 |
| | A2 | 100 | 2,297 | 0.89 | 0.90 | 0.68 | 0.75 | 0.63 | 0.74 |
| | validated | 170 | 4,010 | 0.85 | 0.87 | 0.64 | 0.73 | 0.59 | 0.71 |

Table 4: Scores for each pre-annotation method.

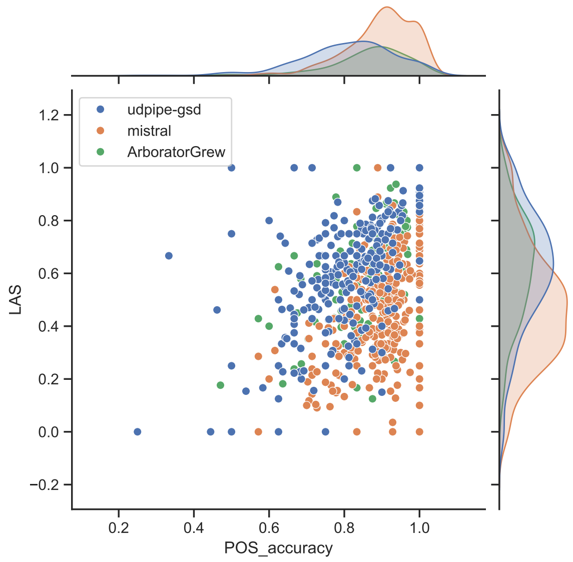


Figure 6: Per sentence POS accuracy and LAS for UDPipe-GSD, Mistral and ArboratorGrew with kernel density estimate (KDE) plots.

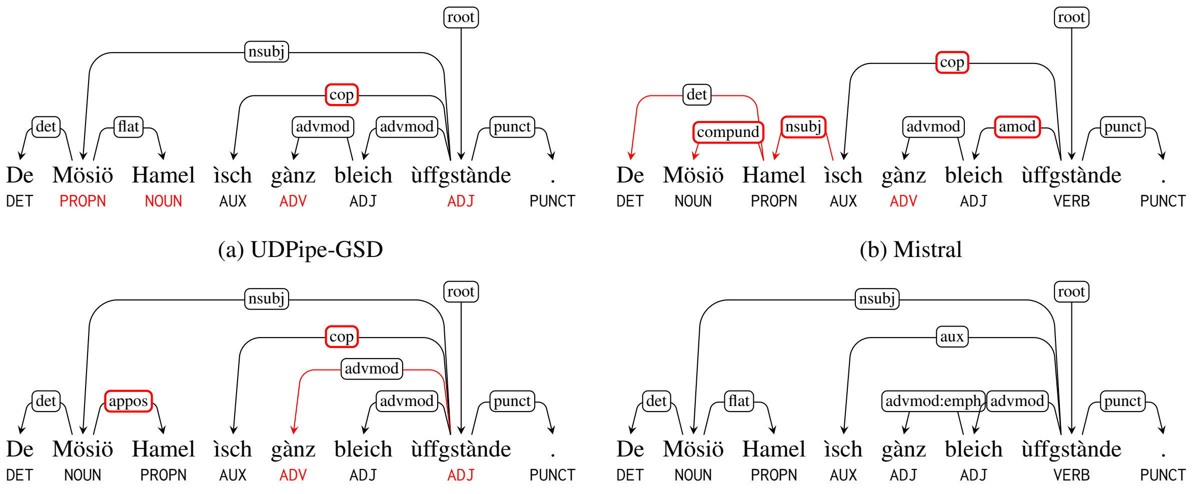
higher concentration of sentences with higher POS accuracy, but lower LAS. UDPipe-GSD and AboratorGrew have a higher concentration of points towards the top half LAS values.

If we compare mean dependency distances[12] across the same sentences, Mistral is characterized by shorter distances (avg=3.05, median=3.17), while UDPipe-GSD has larger distances (avg=3.40, median=3.47) closer to what is observed in the validated sentences (avg=3.41, median=3.59), showing that dependency analyses by Mistral tend to favour connections with less intervening words.

Figure 7 compares the pre-annotations of a sentence against the version validated by the annotators. The pre-annotations from UDPipe-GSD, Mistral and ArboratorGrew contain errors in both POS tags and dependencies. While all three pre-annotation tools correctly identified the root of the sentence, all three mistook the perfect tense as a copular structure. The noun phrase "De Mösiö Hamel" (*Mister Hamel*) was correctly identified as the subject of the sentence by all three tools, but both the internal structure and the POS of the elements was a source of error. It is also interesting to note that all three tools annotated the word "gànz" as an adverb (both in POS and for its dependency), whereas the annotators followed annotation guidelines and annotated this word with the POS 'ADJ', although it functions as an adverb. This example shows that there are different types of errors between different pre-annotations: UDPipe-GSD performed worst for POS tags, but best for dependencies, with only one error. On the contrary, Mistral performed best for POS tags, but lower for dependencies. ArboratorGrew lies in between.

For **RQ2** (*Which pre-annotation method is the most useful?*), we find that there are notable differences among the pre-annotation methods, according to the task: simpler POS or dependency labelling tasks can be performed in-context with an instruction-tuned LLM; however more complex dependency attachment resolution is better achieved by models specifically trained for dependency parsing. The best compromise between both tasks is achieved by ArboratorGrew: the model has been trained on comparatively less data than both UDPipe models (2,192 sentences vs. 13,814 sentences in GSD-train and 153,035 sentences in HDT-train), but on a mix of closely related languages and dialects, with variation in writing characteristic of dialects. This is in line with Philippy et al. (2023)

[12] Calculated by averaging the absolute distance between a word and its head, excluding the root (Liu et al., 2017).


root
nsubj
cop
det
flat
advmod
advmod
punct
De Mösiö Hamel ìsch gànz bleich ùffgstànde .
DET PROPN NOUN AUX ADV ADJ ADJ PUNCT
root
cop
det
compund
nsubj
advmod
amod
punct
De Mösiö Hamel ìsch gànz bleich ùffgstànde .
DET NOUN PROPN AUX ADV ADJ VERB PUNCT
nsubj
root
cop
advmod
det
appos
advmod
punct
De Mösiö Hamel ìsch gànz bleich ùffgstànde .
DET NOUN PROPN AUX ADV ADJ ADJ PUNCT
nsubj
root
aux
det
flat
advmod:emph
advmod
punct
De Mösiö Hamel ìsch gànz bleich ùffgstànde .
DET NOUN PROPN AUX ADJ ADJ VERB PUNCT

(a) UDPipe-GSD

(b) Mistral

(c) ArboratorGrew

(d) Version validated by the annotators

Figure 7: Comparison of the pre-annotations with the validated version for the sentence "De Mösiö Hamel ìsch gànz bleich ùffgstànde" – *'Mister Hamel stood up all pale'*. Errors are marked in red.

who show that cross-lingual transferability is linked to linguistic similarity. It also confirms observations by Blaschke et al. (2024) who obtained lower results for Bavarian with HDT than GSD, despite its larger training corpus: this could be due to an over-fitting of the HDT model for standard German, or to larger discrepancies in terms of genres and domains between the HDT corpus and the Bavarian and Alsatian corpora.

### 4.3 Pre-annotation Bias

Table 5 shows the correlations between the proportion of tokens pre-annotated by a tool and the global agreement of the annotators with the pre-annotation in a batch. The significant correlation scores show that there is a negative correlation for POS pre-annotation by UDPipe-HDT: the higher the proportion of tokens pre-annotated by UDPipe-HDT, the lower the agreement between the annotators and the POS pre-annotation. This means that the annotators tended to correct and modify the POS pre-annotations by UDPipe-HDT. On the other-hand, there is a positive correlation for dependency pre-annotation for UDPipe-GSD and, to a lesser degree UDPipe-HDT. The observations are in line with the performances of the systems shown in Table 4. Higher agreements with the pre-annotations for dependencies are observed when there is a higher proportion of the best performing tools among the pre-annotations and lower agreements with the POS pre-annotations occur when there is a higher proportion of the lowest performing system. This shows that the annotators were able to identify good and low-quality pre-annotations and tended to agree with correct pre-annotations.

| Score | Pre-annotation | Spearman | Pearson |
|---|---|---|---|
| POS | UDPipe-GSD | $-0.50$ | $-0.30$ |
| | UDPipe-HDT | $-0.82^{**}$ | $-0.71^{*}$ |
| | Mistral | $0.43$ | $0.42$ |
| | ArboratorGrew | $0.89^{*}$ | $0.68$ |
| Dep | UDPipe-GSD | $0.84^{***}$ | $0.90^{***}$ |
| | UDPipe-HDT | $0.79^{**}$ | $0.89^{**}$ |
| | Mistral | $-0.57$ | $-0.66^{*}$ |
| | ArboratorGrew | $-0.37$ | $-0.38$ |

Table 5: Spearman's and Pearson's correlations between the proportion of tokens pre-annotated by a tool and the agreement between the annotators and the pre-annotation in a batch. $P$-values: $^{***} < 0.001$, $^{**} < 0.01$ and $^{*} < 0.05$.

Table 4 additionally shows that both A1 and A2 have similar patterns of agreement with the pre-annotation methods, and this agreement is dependent both on the pre-annotation and the task. For **RQ3** (*Can pre-annotation bias be mitigated by using a mix of pre-annotation tools or, on the contrary, does it have a detrimental effect on annotation quality?*), we observe that the annotators did not approach pre-annotations indiscriminately, but rather adapted their correction efforts to the pre-annotation, without uncritically accepting it. Diverse pre-annotation methods thus lead to different correction strategies.

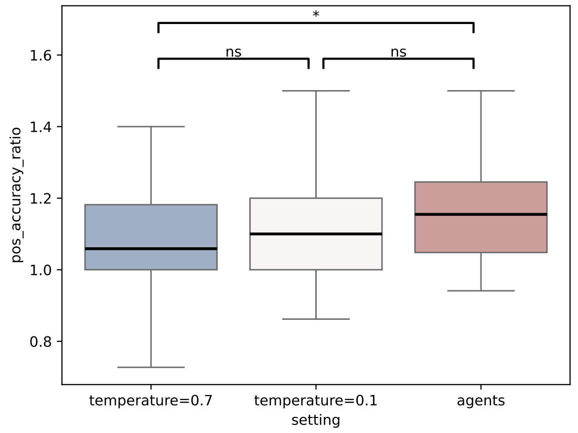


Figure 8: Sentence-level POS accuracy ratio of Mistral in different settings with respect to UDPipe-GSD. Outliers are not shown.

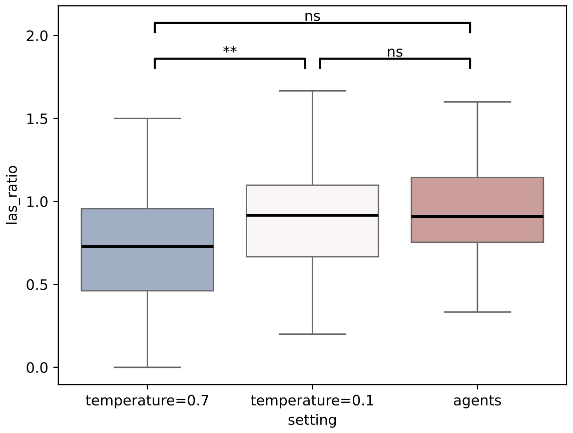


Figure 9: Sentence-level LAS ratio of Mistral in different settings with respect to UDPipe-GSD. Outliers are not shown.

### 4.4 Instruction-tuned LLMs for Pre-Annotation

Since the way we used Mistral evolved in the course of the annotation period, we perform a detailed analysis of Mistral settings (temperatures and agents) in comparison to UDPipe-GSD. For this, we compute sentence-wise ratios of Mistral over UDPipe-GSD for POS accuracy and LAS. By calculating these ratios sentence-wise, we control for the input sentences and their complexity.

Figure 8 shows the distribution of the POS accuracy ratios. These ratios have a median greater than 1, showing that Mistral performs better than UDPipe-GSD for POS tagging. The statistical significance of the difference between the different settings has been assessed using Mann-Whitney's U test (Mann and Whitney, 1947). Only the difference between the temperature of 0.7 and the use of agents is significant. This might indicate that breaking down a complex task into smaller, simpler tasks (here, using agents) can be beneficial.

Figure 9 shows the distribution of the LAS ratios. These ratios have a median inferior to 1, showing that Mistral performs worse than UDPipe-GSD for dependency parsing. Here, only the difference between both temperature settings is significant, with better performance for a temperature of 0.1. Overall, the settings with a higher temperature have the lowest performance: data annotation is not a creative task and it makes sense to set the temperature to its lowest possible value and keep only the most plausible annotation (Gilardi et al., 2023). For **RQ4** (*What are the advantages and pitfalls of instruction-tuned LLMs for our target tasks?*), we find that Mistral is most efficient for simpler labelling tasks at lower temperatures. Besides, as already mentioned, we had to post-process the output to obtain valid CoNLL-U files, which is a clear downside of this method.

## 5 Conclusion and Perspectives

In this work, we have compared three pre-annotation methods for POS and dependency annotation for Low Alemannic Alsatian. Since there is no pre-existing annotated corpus for the language, we used mostly zero-shot methods, relying on closely-related languages or an instruction-tuned LLM. We were able to obtain good annotation quality and showed that the human annotators adapted their correction effort to the perceived quality of the pre-annotation. Moreover, the best method for pre-annotation is task-dependent, with the ArboratorGrew model trained on a mixture of closely-related languages and dialects achieving the best overall performance for both tasks.

The corpus described in this paper is currently being reviewed for its release on the UD repository and will complement the resources already available for High German languages. We also used this corpus to train a parser specifically for Alsatian and pre-annotate a second corpus of texts natively written in Alsatian.

## Limitations

**Selection of pre-annotations for each sentence.** The comparison of the pre-annotation systems does not rely on the exact same set of sentences for each system, since different pre-annotations were used for each sentence and human annotator. Therefore, we could not compare the methods on an identical sample of data. It is therefore possible that the random pre-annotation selection process was more advantageous for some systems (shorter and less complex sentences).

**Pre-annotation methods.** We only compared a restricted set of pre-annotation methods. For the instruction-tuned LLM, only Mistral Large was used, with a single type of prompt. The conclusions could therefore be different for another LLM or for other prompting schemes. Moreover, the pre-annotation tools were used out-of-the-box, without any attempt at tuning the hyperparameters.

**Settings for the pre-annotation systems.** The settings used for some of the pre-annotation systems (UDPipe training corpus version, Mistral prompt) evolved slightly in the course of the four month annotation period, which could impact the consistency of the observations.

**Corpus and language.** The corpus under study includes only one target language and it is unclear how our conclusions could be extended to other languages.

## Acknowledgments

This work has been carried out within the framework of the ANR-21-CE27-0004 DIVITAL project supported by the French National Research Agency. We would like to thank Adrien Fernique and Carole Werner for translating the texts and Michel Feltin-Palas and Daniel Steiner for allowing us to use their content. We also thank Justine Binot for elaborating the initial Mistral Prompt and Carole Werner for training the ArboratorGrew model.

## References

Noëmi Aepli. 2018. Parsing Approaches for Swiss German. Master's thesis, University of Zurich.

Noëmi Aepli and Rico Sennrich. 2022. Improving Zero-Shot Cross-lingual Transfer Between Closely Related Languages by Injecting Character-Level Noise. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4074–4083, Dublin, Ireland. Association for Computational Linguistics.

Delphine Bernhard. 2023. Transfert zero-shot pour l'étiquetage morphosyntaxique : analyse de l'impact de la transformation des données à étiqueter pour les dialectes alsaciens. In *Actes des 5èmes journées du Groupement de Recherche CNRS " Linguistique Informatique, Formelle et de Terrain "*, pages 30–38, Nancy, France.

Delphine Bernhard, Pascale Erhart, Dominique Huck, and Lucie Steiblé. 2018. Part-of-speech annotation guidelines for the alsatian dialects. Zenodo: 10.5281/zenodo.1171925.

Delphine Bernhard and Anne-Laure Ligozat. 2013. Hassle-free POS-Tagging for the Alsatian Dialects. In Marcos Zampieri and Sascha Diwersy, editors, *Non-Standard Data Sources in Corpus Based-Research*, ZSM Studien, pages 85–92. Shaker.

Yevgeni Berzak, Yan Huang, Andrei Barbu, Anna Korhonen, and Boris Katz. 2016. Anchoring and Agreement in Syntactic Annotations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2215–2224, Austin, Texas. Association for Computational Linguistics.

Verena Blaschke, Barbara Kovačić, Siyao Peng, Hinrich Schütze, and Barbara Plank. 2024. MaiBaam: A Multi-Dialectal Bavarian Universal Dependency Treebank. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*.

Verena Blaschke, Hinrich Schütze, and Barbara Plank. 2023. Does Manipulating Tokenization Aid Cross-Lingual Transfer? A Study on POS Tagging for Non-Standardized Languages. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 40–54, Dubrovnik, Croatia. Association for Computational Linguistics.

Emanuel Borges Völker, Maximilian Wendt, Felix Hennig, and Arne Köhn. 2019. HDT-UD: A very large Universal Dependencies Treebank for German. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 46–57, Paris, France. Association for Computational Linguistics.

Philippe Boula de Mareüil, Frédéric Vernier, and Albert Rilliard. 2018. A Speaking Atlas of the Regional Languages of France. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, page 6, Miyazaki, Japan.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Marie-Catherine De Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics*, 47(2):255–308.

Wietse de Vries, Martijn Wieling, and Malvina Nissim. 2022. Make the Best of Cross-lingual Transfer: Evidence from POS Tagging with over 100 Languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7676–7685, Dublin, Ireland. Association for Computational Linguistics.

Stefanie Dipper, Cora Haiber, Anna Maria Schröter, Alexandra Wiemann, and Maike Brinkschulte. 2024. Universal Dependencies: Extensions for Modern and Historical German. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17101–17111, Torino, Italia. ELRA and ICCL.

Timothy Dozat and Christopher D Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *Proceedings of ICLR 2017*.

Hanne Martine Eckhoff and Aleksandrs Berdičevskis. 2016. Automatic parsing as an efficient pre-annotation tool for historical texts. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 62–70, Osaka, Japan. The COLING 2016 Organizing Committee.

Jason M. Eisner. 1996. Three New Probabilistic Models for Dependency Parsing: An Exploration. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Ana Ezquerro, Carlos Gómez-Rodríguez, and David Vilares. 2025. Better benchmarking LLMs for zero-shot dependency parsing. In *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pages 121–135, Tallinn, Estonia. University of Tartu Library.

Karën Fort and Benoît Sagot. 2010. Influence of pre-annotation on POS-tagged corpus development. In *Proceedings of the Fourth ACL Linguistic Annotation Workshop*, pages 56–63.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences of the United States of America*, 120(30):1–3.

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 5293–5302, Marseille, France. European Language Resources Association.

Kirian Guiller. 2020. Analyse syntaxique automatique du pidgin-créole du Nigeria à l'aide d'un transformer (BERT) : Méthodes et Résultats. Master's thesis, Sorbonne Nouvelle - Paris 3.

Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2024. Glottolog 5.1.

Jirka Hana, Anna Feldman, and Katsiaryna Aharodnik. 2011. A Low-budget Tagger for Old Czech. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH '11)*, pages 10–18.

John D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Klaus Krippendorff. 1970. Estimating the reliability, systematic error and random error of interval data. *Educational and psychological measurement*, 30(1):61–70.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of Life Reviews*, 21:171–193.

Andy Lücking, Giuseppe Abrami, Leon Hammerla, Marc Rahn, Daniel Baumartz, Steffen Eger, and Alexander Mehler. 2024. Dependencies over Times and Tools (DoTT). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4641–4653, Torino, Italia. ELRA and ICCL.

Henry B Mann and Donald R Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, and 1 others. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97.

Marie Mikulová, Milan Straka, Jan Štěpánek, Barbora Štěpánková, and Jan Hajic. 2022. Quality and Efficiency of Manual Annotation: Pre-annotation Bias. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2909–2918, Marseille, France. European Language Resources Association.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-Based Dependency Parsing. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 49–56, Boston, Massachusetts, USA. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(null):2825–2830.

Ziqian Peng, Kim Gerdes, and Kirian Guiller. 2022. Pull your treebank up by its own bootstraps. In *Actes Des Journées Jointes Des Groupements de Recherche Linguistique Informatique, Formelle et de Terrain (LIFT) et Traitement Automatique Des Langues (TAL).*, pages 139–153, Marseille, France.

Fred Philippy, Siwen Guo, and Shohreh Haddadan. 2023. Towards a Common Understanding of Contributing Factors for Cross-Lingual Transfer in Multilingual Language Models: A Review. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5877–5891, Toronto, Canada. Association for Computational Linguistics.

Alistair Plum, Caroline Döhmer, Emilia Milano, Anne-Marie Lutgen, and Christoph Purschke. 2024. LuxBank: The First Universal Dependency Treebank for Luxembourgish. In *Proceedings of the 22nd Workshop on Treebanks and Linguistic Theories (TLT 2024)*, pages 30–39, Hamburg,Germany. Association for Computational Linguistics.

Arne Skjærholt. 2014. A chance-corrected measure of inter-annotator agreement for syntax. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 934–944, Baltimore, Maryland. Association for Computational Linguistics.

Daniel Steiner and Raymond Matzen. 2016. *D'Biwel uf Elsässisch*. Éditions du Signe, Strasbourg.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

The pandas development team. 2024. pandas-dev/pandas: Pandas v2.2.3.

Zeno Vandenbulcke, Lukas Vermeire, and Miryam de Lhoneux. 2024. Recipe for Zero-shot POS Tagging: Is It Useful in Realistic Scenarios? In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 137–147, Miami, Florida, USA. Association for Computational Linguistics.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 16 others. 2020. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding Pretrained Models to Thousands More Languages via Lexicon-based Adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Michael L. Waskom. 2021. Seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

## A Outline of the Mistral Prompt

Please note that the details about UD POS tags and dependency relationships, as well as the description of the CoNLL-U format have been removed as these can be found on the Universal Depedencies website. The prompt was elaborated and optimised during earlier experiments with instruction-tuned LLMs, based on commonly acknowledged recommendations for prompting: defining the context and the role of the system, task description and constraints, addition of an example with the expected result, use of delimiters to identify subparts of the prompt.

```
You are an expert in Alsatian annotation. Your
task is to add the missing part-of-speech and
dependencies annotations to the Alsatian
sentences.

Here is the list of UPOS labels to use for
part-of-speech annotations:
<List of UPOS labels with names>

Here is the list of labels for Universal
Dependencies:
<List of relations with names>

Constraints: The output must respect the format
called CoNLL-U. Annotations are encoded in plain
text files (UTF-8, normalized to NFC, using only
the LF character as line break, including an LF
character at the end of file) with three types of
lines:
1.  Word lines containing the annotation of
a word/token/node in 10 fields separated by single
tab characters; see below.
2.  Blank lines marking sentence boundaries. The
last line of each sentence is a blank line.
3.  Sentence-level comments starting with hash (#).
Comment lines occur at the beginning of sentences,
before word lines.
Sentences consist of one or more word lines, and
word lines contain the following fields:
<List of fields in a CoNLL-U file>
The fields must additionally meet the following
```

```
constraints:
• Fields must not be empty.
• Fields other than FORM, LEMMA, and MISC must
not contain space characters.
• Underscore (_) is used to denote unspecified
values in all fields except ID.
Further, in UD treebanks the UPOS, HEAD, and
DEPREL columns are not allowed to be left
unspecified except in multiword tokens, where all
must be unspecified, and empty nodes, where UPOS
is optional and HEAD and DEPREL must be
unspecified.

####
Here is an example:

Sentence:
# sent_id = WKP_12043.19
# text = Isch dr Hans Baldung im Elsàss uf d Walt
    kumme?
1 Isch  _ _ _ _ _ _ _ _
2 dr  _ _ _ _ _ _ _ _
3 Hans  _ _ _ _ _ _ _ _
4 Baldung _ _ _ _ _ _ _ _
5-6 im  _ _ _ _ _ _ _ _
5 i _ _ _ _ _ _ _ SpaceAfter=No
6 m _ _ _ _ _ _ _ _
7 Elsàss  _ _ _ _ _ _ _ _
8 uf  _ _ _ _ _ _ _ _
9 d _ _ _ _ _ _ _ _
10  Walt  _ _ _ _ _ _ _ _
11  kumme _ _ _ _ _ _ _ SpaceAfter=No
12  ? _ _ _ _ _ _ _ _

Annotation:
# sent_id = WKP_12043.19
# text = Isch dr Hans Baldung im Elsàss uf d Walt
    kumme?
1 Isch  _ AUX _ _ 11  aux _ Gloss=est
2 dr  _ DET _ _ 3 det _ Gloss=le
3 Hans  _ PROPN _ _ 11  nsubj _ Gloss=Hans
4 Baldung _ PROPN _ _ 3 flat:name _ Gloss=Baldung
5-6 im  _ _ _ _ _ _ _ _
5 i _ ADP _ _ 7 case  _ SpaceAfter=No|Gloss=dans
6 m _ DET _ _ 7 det _ Gloss=le
7 Elsàss  _ PROPN _ _ 11  obl:lmod  _ Gloss=Alsace
8 uf  _ ADP _ _ 10  case  _ Gloss=en
9 d _ DET _ _ 10  det _ Gloss=le
10  Walt  _ NOUN  _ _ 11  obl _ Gloss=monde
11  kumme _ VERB  _ _ 0 root  _ SpaceAfter=No|Gloss=
    venir
12  ? _ PUNCT _ _ 11  punct _ Gloss=.
####

### Step 1: You must read and understand the
    Alsatian sentences.
### Step 2: Use your understanding from step 1 to
    add the POS, dependency and head labels
### Step 3: Provide the annotation of the given
    sentences.
The annotation should be in the CoNLL-U format.
Your output should consist exclusively of the
annotations. No other comments or text should be
included. Remove markdown formatting.
```

## B Libraries Used

The following Python libraries were used for performing the analyses and drawing the plots:

- conllu v. 6.0.0 (https://github.com/EmilStenstrom/conllu/)
- matplotlib v. 3.9.4 (Hunter, 2007)
- pandas v. 2.2.3 (The pandas development team, 2024)
- scikit-learn v. 1.6.1 (Pedregosa et al., 2011)
- scipy v. 1.13.1 (Virtanen et al., 2020)
- seaborn v. 0.13.2 (Waskom, 2021)
- starbars v. 3.1.1 (https://github.com/elide-b/starbars)

## C Models Used

The following models were used:

- UDPipe:
  - GSD 2.12 and 2.15
  - HDT 2.12 and 2.15
- Mistral Large latest (the latest available Mistral model was always used):
  - unique prompt with temperatures 0.1 and 0.7
  - agents: 4 distinct agents all used in a row with temperature 0
    - UPOS: UPOS annotations
    - Gloss: French glosses
    - Dependencies: dependency annotations
    - CoNLL-U format checker

## D Detailed Scores per Batch

| Batch | Annot. 1 | Annot. 2 | Kappa POS | Acc POS | Alpha Dep | UAS | LAS | LAcc |
|---|---|---|---|---|---|---|---|---|
| 1 | A1 | validated | 0.94 | 0.94 | 0.95 | 0.92 | 0.85 | 0.90 |
| | | pre-annotation | 0.86 | 0.88 | 0.75 | 0.67 | 0.58 | 0.72 |
| | A2 | validated | 0.96 | 0.96 | 0.96 | 0.93 | 0.88 | 0.92 |
| | | pre-annotation | 0.84 | 0.86 | 0.78 | 0.70 | 0.60 | 0.72 |
| | A1 | A2 | 0.90 | 0.91 | 0.88 | 0.87 | 0.77 | 0.84 |
| 2 | A1 | validated | 0.95 | 0.95 | 0.97 | 0.94 | 0.89 | 0.93 |
| | | pre-annotation | 0.85 | 0.86 | 0.70 | 0.65 | 0.55 | 0.71 |
| | A2 | validated | 0.96 | 0.96 | 0.98 | 0.96 | 0.91 | 0.93 |
| | | pre-annotation | 0.87 | 0.88 | 0.63 | 0.61 | 0.51 | 0.71 |
| | A1 | A2 | 0.91 | 0.92 | 0.94 | 0.92 | 0.82 | 0.87 |
| 3 | A1 | validated | 0.95 | 0.95 | 0.97 | 0.93 | 0.88 | 0.92 |
| | | pre-annotation | 0.86 | 0.87 | 0.71 | 0.65 | 0.53 | 0.69 |
| | A2 | validated | 0.95 | 0.96 | 0.98 | 0.94 | 0.90 | 0.94 |
| | | pre-annotation | 0.87 | 0.88 | 0.64 | 0.66 | 0.55 | 0.73 |
| | A1 | A2 | 0.91 | 0.92 | 0.94 | 0.89 | 0.81 | 0.88 |
| 4 | A1 | validated | 0.94 | 0.95 | 0.98 | 0.94 | 0.88 | 0.92 |
| | | pre-annotation | 0.90 | 0.91 | 0.70 | 0.73 | 0.63 | 0.76 |
| | A2 | validated | 0.95 | 0.96 | 0.97 | 0.95 | 0.91 | 0.94 |
| | | pre-annotation | 0.86 | 0.87 | 0.77 | 0.75 | 0.61 | 0.73 |
| | A1 | A2 | 0.91 | 0.92 | 0.94 | 0.90 | 0.81 | 0.87 |
| 5 | A1 | validated | 0.94 | 0.94 | 0.98 | 0.94 | 0.91 | 0.95 |
| | | pre-annotation | 0.86 | 0.88 | 0.71 | 0.70 | 0.57 | 0.70 |
| | A2 | validated | 0.97 | 0.98 | 0.98 | 0.96 | 0.93 | 0.95 |
| | | pre-annotation | 0.87 | 0.88 | 0.64 | 0.69 | 0.58 | 0.72 |
| | A1 | A2 | 0.92 | 0.92 | 0.94 | 0.92 | 0.86 | 0.91 |
| 6 | A1 | validated | 0.97 | 0.98 | 0.98 | 0.95 | 0.91 | 0.95 |
| | | pre-annotation | 0.83 | 0.84 | 0.69 | 0.67 | 0.55 | 0.69 |
| | A2 | validated | 0.96 | 0.97 | 0.99 | 0.97 | 0.94 | 0.96 |
| | | pre-annotation | 0.87 | 0.89 | 0.65 | 0.64 | 0.52 | 0.68 |
| | A1 | A2 | 0.94 | 0.94 | 0.96 | 0.93 | 0.87 | 0.91 |

Table 6: Detailed scores for each annotation batch.