universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Syntactic Parsing with Generative Language Models: The Impact of In-Context Learning on Romance Minority Languages

verfasst von | submitted by

Marlene Albrecht BA

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Arts (MA)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 647
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Digital Humanities
Betreut von | Supervisor:	Univ.-Prof. Dr. -Ing. Benjamin Roth B.Sc. M.Sc.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Benjamin Roth, for his valuable guidance, constructive feedback, and continuous support throughout the process of writing this master's thesis. His expertise and advice have been highly valuable in completing this work.

My deepest thanks go to Terra Blevins. Her generous support and the time she dedicated, as well as her helpful advice and ideas greatly enriched this thesis. Her encouragement and her willingness to share knowledge have been crucial for the development of this work and have contributed significantly to my academic growth.

Finally, I wish to extend my heartfelt appreciation to my family, my friends, and my partner for their understanding, patience, and constant support throughout this journey.

Abstract

Training generative language models often requires large amounts of data, which are not available for all languages. In-context learning offers an approach in which models learn from examples shown in the prompt. The goal of this research was to investigate how factors such as the number of example annotations, linguistic diversity, language similarity between the source and target languages, and syntactic similarity influence model output. To this end, in-context learning experiments on POS tagging and dependency parsing were conducted with Romance languages and Basque, focusing on the Romance minority languages Galician, Catalan, and Occitan. The results indicated that showing examples hardly affects model performance when structural and linguistic constraints are applied, as even 0-shot results are promising. Additionally, constraints created for high-resource languages also yield satisfactory results for minority languages. Furthermore, in some cases, the linguistic and syntactic similarity of the example annotations correlates with model performance, although no clear relation can be deduced. This research demonstrates the potential of computational linguistic methods in traditional philological disciplines, especially Romance studies, and their contribution to optimizing procedures for underrepresented languages.

Abstract (Deutsch)

Für das Training generativer Sprachmodelle werden oft große Datenmengen benötigt, die allerdings nicht für alle Sprachen verfügbar sind. In-Context Learning bietet einen Ansatz, bei dem Modelle aus den im Prompt gezeigten Beispielen lernen. Das Ziel dieser Forschung war es, zu untersuchen, wie Faktoren wie Anzahl der Beispielannotationen, linguistische Diversität, Sprachähnlichkeit zwischen Quell- und Zielsprache und syntaktische Ähnlichkeit den Modelloutput beeinflussen. Hierfür wurden In-Context Learning Experimente zu POS-Tagging und Dependency Parsing mit romanischen Sprachen und Baskisch durchgeführt, wobei die romanischen Minderheitensprachen Galizisch, Katalanisch und Okzitanisch im Fokus standen. Die Ergebnisse zeigten, dass das Zeigen von Beispielen wenig Einfluss auf die Modellperformanz hat, wenn strukturelle und linguistische Constraints eingesetzt werden, da bereits 0-Shot Ergebnisse überzeugend sind. Darüber hinaus erzielen Constraints, die für eine Hochressourcensprache erstellt wurden, auch für Minderheitensprachen gute Ergebnisse. Zudem korrelieren Sprach- und syntaktische Ähnlichkeit der Beispielannotationen in manchen Fällen mit der Modellperformanz, wobei jedoch keine klaren Zusammenhänge ableitbar sind. Diese Forschung demonstriert das Potenzial computerlinguistischer Methoden in traditionellen philologischen Disziplinen, insbesondere der Romanistik, und deren Beitrag zur Optimierung von Verfahren für unterrepräsentierte Sprachen.

Table of Contents

List of Tables

List of Figures

1. Introduction

Natural language processing (NLP) research underrepresents the Romance minority languages Catalan, Galician, and Occitan because of the limited availability of annotated data. This thesis investigates the ability of generative language models in syntactic parsing and tests whether in-context learning leads to an improvement in model performance, especially regarding the three Romance minority languages.

This is investigated through few-shot experiments in 0-, 1-, 3-, 5-, and 10-shot scenarios, in which the model learns solely from the examples provided in the prompt (Brown et al., 2020). The aim is to investigate the impact of the number of annotations, linguistic diversity, language similarity, and syntactic similarity on parsing performance. In addition to the three minority languages, the high-resource Romance languages French, Spanish, and Portuguese, as well as the language isolate Basque, are also being tested. Combining these variables in a systematic few-shot setup constitutes a novel approach that has not yet been investigated for these languages.

The results provide indications as to whether in-context learning can improve model performance, particularly for low-resource languages where large training datasets are not available. This information is relevant for improving the efficiency of annotations and providing targeted support for low-resource languages. In addition to free generation, whose output can be unstructured, constrained decoding is also used, a technique that structures the output of language models (Geng et al., 2024). We show that constrained decoding not only facilitates the structuring of output but also offers great opportunities for the syntactic parsing of low-resource languages with the help of cross-lingual transfer.

The following chapters present the fundamentals of syntactic parsing and in-context learning (2), the model and data used (3), prompt engineering and in-context learning experiments (4), followed by a presentation of the results (5) and a series of quantitative and qualitative analysis steps of the model output (6). With this study, the thesis aims to stimulate interest in minority languages in the field of NLP and, in addition, contributes to the interdisciplinary connection between computational linguistics and Romance studies.

2. Background

To provide a deeper understanding of the upcoming experiments, this chapter introduces the relevant fundamentals of syntactic parsing, dependency parsing with generative large language models (LLMs), and in-context learning. It also presents the current state of research to enable the placement of the presented approaches in a scientific context. The chapter concludes with the motivation behind this research and the central research question.

2.1 Syntactic Parsing

In the context of this thesis, the term *syntactic parsing* refers to both part-of-speech tagging (POS tagging) and dependency parsing. POS tagging describes the task of assigning part-of-speech labels (also known as word classes or syntactic categories) to words in an input text. The classification of words has a long tradition and most likely dates to ancient times. Nowadays, part-of-speech tagging is one of the most fundamental statistical models for many NLP applications. It serves as a core aspect of parsing and is used to label named entities, extract information, and can also play a role in speech recognition and synthesis (Kumawat & Jain, 2015). Traditional approaches to POS tagging include rule-based methods, statistical methods, and hybrid methods, which combine rule-based and statistical approaches. Rule-based methods are based on linguistic heuristics, in which, for example, word endings or word contexts provide information about which POS tag a word receives. However, one disadvantage of these systems is that they are difficult to scale and require detailed linguistic knowledge. Statistical methods, on the other hand, work with probability models that have been trained on annotated data. The best-known statistical POS tagging methods are the Hidden Markov Model (HMM), the Maximum Entropy Model (MaxEnt), and the Conditional Random Fields Model (CRF). The disadvantage of these models is that they require large annotated corpora for training, which leads to considerable problems, especially for low-resource languages (Jurafsky & Martin, 2025; Kumawat & Jain, 2015; Ratnaparkhi, 1996). While POS tagging involves labeling individual words, dependency parsing focuses on finding syntactic structures within a sentence. Dependency parsing goes back to Lucien Tesnière's dependency grammar and is a process for analyzing grammatical structures in a sentence and determining the dependencies of words on each other (Tesnière, 1959). Dependency relations are binary relations consisting of a head and a dependent. The dependents are either directly or indirectly related to the head. The dependents are also further defined by the description of their grammatical function. Universal dependencies provide a standard for dependency

relations (Nivre, 2020; Nivre et al., 2016). As part of the Universal Dependencies project, an inventory of dependency relations was created that is linguistically sound, computationally useful, and applicable across languages (Jurafsky & Martin, 2025).

Dependency parsing is an important part of natural language processing and is used, for example, for machine translation or question answering. Creating a parser for low-resource languages is often challenging, as the creation of a parser often requires large, annotated treebanks (Zeman & Resnik, 2008). Traditional approaches to dependency parsing include transition-based parsing and graph-based parsing. In the former, the parsing process is modeled as a sequence of actions or transitions to build the dependency tree step by step, either from left to right or from right to left. Deterministic and efficient algorithms such as arc-eager and arc-standard are used here. Transition-based parsers typically have linear time complexity, which makes them suitable for real-time parsing tasks (Jurafsky & Martin, 2025; Nivre, 2003). In graph-based parsing, the most probable tree structure for an input sentence is sought. These parsers are often more computationally complex but are better at capturing dependencies in longer and more complex sentences. The main difference between transition-based and graph-based parsing is that in the former, the parse tree is constructed step by step through a sequence of transitions, while in the latter, all possible parse trees are considered as graphs and the optimal spanning tree is sought (Bohnet, 2016; Jurafsky & Martin, 2025). While transition-based parsing is generally more efficient, graph-based parsing can achieve higher accuracy, especially for complex sentence structures. Since both approaches have different strengths, the two methods have also been combined, for example in Y. Zhang and Clark (2008). Well-known dependency parsers include the transition-based SpaCy DependencyParser[1], the Stanza Parser[2] (Qi et al., 2020) and StanfordCoreNLP[3] (Manning et al., 2014).

The most common methods for evaluating dependency parsers are labeled attachment accuracy and unlabeled attachment accuracy, commonly referred to as labeled attachment score (LAS) and unlabeled attachment score (UAS). LAS refers to the correct assignment of a word to its head along with the correct dependency relation (DEPREL). In UAS, the assignment to the correct dependency relation is irrelevant, as only the correctness of the assigned head is considered. The system output is compared to the gold standard (reference), and the accuracy (AligndAcc) measures the percentage of words in an output sentence that have been correctly

[1] https://spacy.io/api/dependencyparser

[2] https://stanfordnlp.github.io/stanza/depparse.html

[3] https://stanfordnlp.github.io/CoreNLP/

analyzed. For UAS, this refers to whether the word was assigned to the correct head, and for LAS, whether the word was assigned to the correct head and the correct relation (the correct label). While precision and recall are used to evaluate the parser's performance on specific relation types, accuracy measures the parser's overall performance across a sentence (Jurafsky & Martin, 2025).

Having outlined the basics of dependency parsing and its traditional methods, in the following chapter we will present a newer approach, namely dependency parsing with generative language models.

2.2 Dependency Parsing with Generative LLMs

Even though traditional dependency parsing methods are still widely used, the potential of generative language models for dependency parsing is now being increasingly explored, opening up new approaches in this area of NLP. An important development in this area was the U-DepPLLaMA model by Hromei et al. (2024). The aim of this development was to investigate the use of LLMs (in this case the LLaMA family of models) for end-to-end dependency parsing without task-specific architecture. Dependency parsing is treated as a sequence-to-sequence problem, in which the syntax of a sentence is represented in a bracket structure. Instead of generating an explicit tree structure, the model converts the sentences into annotations in bracket notation. However, this representation can be directly converted into a dependency tree. The model was trained and tested using 50 treebanks in 26 languages from Universal Dependencies (Nivre, 2020; Nivre et al., 2016) and evaluated using the Unlabeled Attachment Score (UAS) and the Labeled Attachment Score (LAS). In a direct comparison with UDPipe2.0 (Straka, 2018), a traditional parsing system, U-DepPLLaMA delivered comparable or even better results. Nevertheless, the model also exhibited some weaknesses or parsing errors. On the one hand, it performed less well in morphologically rich languages such as Korean, Chinese, or Finnish. In addition, the model had problems parsing long sentence dependencies. The most common error occurred with nouns, where the correct labels were assigned, but the wrong dependency was assigned. On the other hand, punctuation marks and root relations were often assigned incorrectly, which can affect the entire sentence analysis. In conclusion, the authors identified the adaptation of U-DepPLLaMA for low-resource languages and the exploration of cross-lingual learning techniques as potential areas of research (Hromei et al., 2024).

Lin et al. (2022) have already proven that dependency parsing is possible with pre-trained language models, as encoder-decoder models are capable of generating relationship structures directly from input texts. Traditional dependency parsers typically work with embeddings and use additional layers to generate predictions. The issue with this is that enormous computing

power is required for parsing, as the method is based on text generation and the calculation time increases significantly with the number of tokens (Kim & Ko, 2025). Therefore, Kim and Ko (2025) presented SPT-DP (Structuralized Prompt Template-Based Dependency Parsing), a new approach to dependency parsing based exclusively on pre-trained encoder models, such as BERT[4] (Devlin et al., 2019) or RoBERTa[5] (Y. Liu et al., 2019). Instead of using traditional methods that require additional layers or modules for predictions, this approach relies on prompt engineering to convert the task into a text-to-text transformation that the pre-trained model can process directly. The authors report that this method outperforms "most existing approaches" and works faster. In tests with 12 languages, the model demonstrated the ability to perform multilingual dependency parsing within a single system and even transfer its parsing capabilities to unfamiliar languages.

Lin et al. (2023) investigated the capabilities of ChatGPT[6] as a zero-shot parser (i.e., without special training and without showing examples), simply by showing a text prompt. It was shown that ChatGPT showed promising results as a zero-shot dependency parser. Cross-lingual experiments showed that ChatGPT also behaves consistently across different languages, and the output retains a similar structure for similar sentences. It was also found that ChatGPT can even detect and avoid errors in official datasets, which offers an advantage over labeled data.

The zero-shot capabilities of ChatGPT as a dependency parser described above illustrate that LMs can perform complex tasks without explicit training, only with a well-formulated prompt. This raises the question of whether this performance can be further enhanced by showing examples in the prompt. This is where the concept of in-context learning comes in, describing how LMs learn new task structures through examples or instructions within the context (Brown et al., 2020). The following chapter therefore presents the theoretical foundations of in-context learning, one of the fundamental concepts for the upcoming experiments.

2.3 In-Context Learning

Recently, the field of natural language processing (NLP) has been revolutionized by the emergence of transformer models such as GPT (Radford et al., 2018) and BERT (Devlin et al., 2019). It has been shown that unsupervised training on large text corpora can improve the performance of the model in numerous applications such as machine translation, text summarization, and

[4] https://huggingface.co/docs/transformers/model_doc/bert

[5] https://huggingface.co/docs/transformers/model_doc/roberta

[6] https://chatgpt.com/

question answering (Y. Liu et al., 2019; Radford et al., 2019; Yang et al., 2020). The need for task-specific architecture could be eliminated by fine-tuning, i.e., adjusting the model weights to the tasks (Howard & Ruder, 2018; Radford et al., 2018). While the paradigm of pretraining followed by specific fine-tuning on downstream tasks has shown significant improvements, it also presents a major limitation, as fine-tuning requires large, annotated datasets for each specific task. In-context learning represents an alternative approach that aims to overcome these limitations. In in-context learning, the pre-trained model learns solely by providing examples in the input context, without the need for adjustment by model parameters. This contrasts with classic fine-tuning, which requires adjustment to each new task. Brown et al. (2020) distinguish between three forms of in-context learning.

- Zero-shot learning (0-shot learning): The model is given a natural language instruction (via a prompt) and is asked to solve the described task directly, without being shown any examples. This is one of the most challenging settings, as even humans might find it difficult to understand a task without being shown an example. Nevertheless, this is also a very convenient and robust method that avoids unwanted correlations (Brown et al., 2020).
- One-shot learning: In one-shot learning (also 1-shot learning), the pre-trained model is given an example as a demonstration in addition to the linguistic instruction. This demonstration of the task, which the model receives at inference time, serves as conditioning (Radford et al., 2019). Weight updates are not allowed. The example shown provides information about the desired output of the model (Brown et al., 2020).
- Few-shot learning: Few-shot learning works in the same manner as one-shot learning, except the model is presented with multiple examples illustrating the task solution rather than just one. The reduction in the need for task-specific data is a significant advantage of few-shot learning. Few-shot learning also reduces the risk of the model only learning specific patterns in the dataset, which can lead to overfitting. One disadvantage, though, is that state-of-the-art fine-tuned models often still show better performance (Brown et al., 2020).

When evaluating generative language models, it became apparent that few-shot learning can achieve a high degree of accuracy without additional training. Another advantage of few-shot learning is that, due to its ability to generalize, it does not require any intermediate storage for what has been learned. However, a key disadvantage is the inference speed, because few-shot learning requires more computing resources per inference step, which usually increase with the

number of examples. The long inference time is due to the fact that the entire context must be reprocessed with each run. This can lead to a bottleneck and often makes real-time application difficult. To further improve the efficiency of in-context learning, a reduction in inference time would be desirable. As already mentioned, in-context learning is particularly suitable when there is limited data available. Therefore, in-context learning is ideal for working with low-resource languages for which only limited annotated gold standard data is available (Brown et al., 2020; Winata et al., 2021).

Winata et al. (2021) tested few-shot learning in a multilingual setting for English, French, German, and Spanish on natural language understanding intent prediction tasks, using usable LMs that were primarily trained on English. The multilingual skills of GPT and T5[7] models were tested in performing multi-class classification without parameter updates. It was found that in-context few-shot cross-lingual prediction delivers significantly better results than random prediction, and the results are comparable to existing state-of-the-art cross-lingual models and translation models.

Having outlined the fundamental concepts such as syntactic parsing and in-context learning with generative language models and having already briefly discussed their advantages in dealing with limited data, we will now move on to the specific motivation and research question of this thesis in the next chapter.

2.4 Motivation and Research Question

Generative language models take on numerous tasks: they are capable of summarizing texts, generating novel texts, and performing translation work (Brown et al., 2020). Until recently, large, human-annotated datasets such as treebanks were a prerequisite for training language models (De Marneffe et al., 2021). These kinds of resources were primarily available in high-resource languages such as English, French, German, etc. For several minority languages, there is hardly any annotated data available, as the number of speakers of these languages is limited and institutional funding for annotation projects in minority languages is considerably lower (Joshi et al., 2020).

The use of generative language models requires minimal technical knowledge, since tasks can be communicated to the model through short prompts in natural language (H. Dang et al., 2022). This reduces the barrier compared to traditional dependency parsers, such as Stanza[8], which require at least minimal programming knowledge. Experiments by Lin et al. (2023) showed

[7] https://huggingface.co/docs/transformers/model_doc/t5

[8] https://stanfordnlp.github.io/stanza/

that generative language models, such as ChatGPT, are capable of cross-lingual transfer, an important feature for supporting low-resource languages by leveraging resources from higher-resource languages.

The aim of this research is to analyze the ability of generative language models in syntactic parsing and to test whether in-context learning leads to an improvement in model performance. Particular attention is paid to how the model handles Romance minority languages. The target languages are Occitan, Catalan, and Galician as Romance minority languages; French, Spanish, and Portuguese as Romance high-resource languages; and Basque, a language isolate, as a non-Romance language. The source languages for the example annotations are the ones mentioned above, as well as Romanian and Italian in mixed datasets. Experiments are conducted with different language pairs, i.e., combinations of different target languages and source languages, in 0-, 1-, 3-, 5-, or 10-shot scenarios to determine the effect of the number of shots and the cross-lingual prediction capabilities of the model. The aim of this research is to answer the following central research question:

(F1) To what extent do various characteristics of example annotations affect model performance during in-context learning for syntactic parsing?

The following characteristics are:

- quantity of annotations
- linguistic diversity
- language similarity
- syntactic similarity

The following hypotheses were formulated for this purpose, which need to be confirmed or refuted:

(H1) A higher number of examples in the prompts results in higher output quality.

(H2) Prompts that contain example sentences that are syntactically similar to the desired target sentence yield particularly accurate results.

(H3) If the example sentences are written in a language that is similar to the target language, this leads to improved output quality.

This thesis aims to provide insights into the ability of generative language models for dependency parsing without specific training. In addition, the effect of in-context learning prompts on the model will be tested. The development of methods to support annotation in Romance studies, especially for languages with limited resources, represents an important contribution. The objective of this work is to draw attention to the linguistic disadvantages of

minority languages and to raise awareness of their value. At the same time, it aims to motivate researchers to engage more intensively with minority languages. It also intends to highlight the interface between computational linguistics and Romance studies. Promoting appropriate computer-assisted annotation techniques, for instance through generative language models, enables researchers to work more efficiently, improve the quality of linguistic resources, and contribute to the preservation and study of languages that have been insufficiently documented to date.

3. Methodology and Data

The following chapter introduces the Aya Expanse 32B language model (J. Dang, Singh, et al., 2024) that was used for the experiments. It then provides information about the datasets used for the experiments and describes how the training sets and development sets are composed. Since no gold standard was available for Occitan, a gold standard was annotated specifically for this project. The process is described in Chapter 3.3.

3.1 Aya-Expanse-32B

The model selected for these experiments is the Aya Expanse 32B[9] (J. Dang, Singh, et al., 2024), developed by Cohere For AI and Cohere. It belongs to a family of instruction-tuned models, an open-weights release with 8 billion (Aya Expanse 8B[10]) and 32 billion parameters (Aya Expanse 32B), which support 23 languages. These languages are Arabic, Chinese (simplified and traditional), Czech, Dutch, English, French, German, Greek, Hebrew, Hindi, Indonesian, Italian, Japanese, Korean, Persian, Polish, Portuguese, Romanian, Russian, Spanish, Turkish, Ukrainian, and Vietnamese. Models perform better in languages they have been trained on, which can lead to bias toward languages the model has not seen. In addition, models require more tokens when processing languages they have not been trained on, which increases latency and thus leads to higher costs when using them. The goal of the Aya Expanse 32B model is to close the language gap (Ahia et al., 2023; J. Dang, Singh, et al., 2024; Khandelwal et al., 2024; Khondaker et al., 2023; Kunchukuttan et al., 2021; Vashishtha et al., 2023; Yong et al., 2024).

The Aya Expanse language model was developed with the aim of matching or exceeding the capabilities of monolingual models as a multilingual language model. Aya Expanse uses a range of innovative technologies in the post-training process, such as multilingual data arbitrage

[9] https://huggingface.co/CohereLabs/aya-expanse-32b

[10] https://huggingface.co/CohereLabs/aya-expanse-8b

(Odumakinde et al., 2024), multilingual preference optimization and safety tuning (Aakanksha, Ahmadian, Ermis, et al., 2024; J. Dang, Ahmadian, et al., 2024) and model merging, which are explained in more detail below (Aakanksha, Ahmadian, Goldfarb-Tarrant, et al., 2024). Training a language model requires a lot of data, which is why current data sources are exhausted. This issue makes the generation of synthetic data increasingly important. Over-reliance on synthetic data can lead to model collapse (Shumailov et al., 2024).

Since there are few good teacher models available for multilingual data, especially for low-resource languages, data arbitrage was used for Aya Expanse. With this technique, synthetic data is generated not only from a single teacher model, but from a pool of models. The models are strategically combined to achieve the best performance for each language. Thanks to this process, Aya Expanse was able to achieve a significant performance improvement compared to previous models even in an early training phase (Odumakinde et al., 2024). Multilingual preference training is used to train a language model to better understand human preferences in multiple languages simultaneously. A multitude of existing preference datasets are only available in English, which is why this technique uses specially generated multilingual training data to adapt the model to different linguistic and cultural contexts. To this end, high-quality in-language completions are compared with poorer translations. This is intended to help the model learn to generate more accurate and natural responses in each language. The process consists of a combination of offline and online preference training: In offline training, existing, verified training data forms a stable basis, while in online training, the model independently generates new responses. In addition, an iterative training process is used, in which a reward model evaluates the responses, and the model is further trained. This strategy enabled Aya Expanse 8B to achieve a 7.1% higher win rate than Gemma 2 9B[11] (Team et al., 2024), underlining its effectiveness (J. Dang, Singh, et al., 2024).

Model merging combines multiple language models to combine their individual strengths and reduce computing costs. In Aya Expanse, model merging is used both during the data arbitrage phase and in each iteration of preference training. Separate models were trained for different language families, with common languages such as English, Spanish, and French serving as connecting elements, with the aim of benefiting from cross-lingual transfer. Various techniques were used for model fusion, with weighted averaging proving to be the best and most consistent method. Aya Expanse 32B demonstrated up to a threefold increase in performance through model merging compared to Aya Expanse 8B (J. Dang, Singh, et al., 2024; Yadav et al., 2024).

[11] https://huggingface.co/google/gemma-2b

To test the effectiveness of the techniques mentioned above, extensive testing was carried out using multilingual benchmarks. The Arena-Hard Auto dataset[12] was used for the main evaluation and translated into 23 languages to enable testing in different linguistic contexts (Li et al., 2024). In addition, the model was also tested on multilingual benchmarks such as m-Arena-Hard[13] and Dolly[14] (Singh et al., 2024), as well as academic benchmarks (J. Dang, Singh, et al., 2024; Team et al., 2024).

Aya Expanse 8B and 32B outperform the leading open-source models in their respective parameter classes, including Llama 3.1[15] (Grattafiori et al., 2024), Qwen 2.5[16] (Qwen et al., 2025), and Gemma 2[17], with a win rate of up to 76.6%. Aya Expanse 32B also beats Llama 3.1 70B[18], which has more than twice as many parameters, with a win rate of 54.0%. The Aya Expanse models stand out from other language models thanks to a post-training strategy. These innovations mean that the models not only perform well in multiple languages but also demonstrate consistent performance across a wide range of tasks. With these technological advances, Aya Expanse sets a new benchmark for multilingual language models (J. Dang, Singh, et al., 2024).

Having introduced the language model, we now present the datasets used in our experiments.

3.2 Data

The experiments were conducted for the target languages Galician, Catalan, and Occitan, as well as French, Spanish, and Portuguese, and as a comparison with the non-Romance language Basque. Although Catalan, Galician, and Occitan are all considered minority languages (Calin, 2001; Davidson, 2022; Sousa, 2020), their linguistic resources in the field of NLP differ considerably. We used established treebanks from Universal Dependencies[19] version 2.14 to create the datasets for this research.

[12] https://huggingface.co/datasets/lmarena-ai/arena-hard-auto

[13] https://github.com/mivanovitch/arena-hard

[14] https://huggingface.co/databricks/dolly-v2-3b

[15] https://huggingface.co/meta-llama/Llama-3.1-8B

[16] https://chat.qwen.ai/

[17] https://huggingface.co/google/gemma-2-2b

[18] https://huggingface.co/meta-llama/Llama-3.1-70B

[19] https://universaldependencies.org

3.2.1 Catalan

Despite its minority status, Catalan is considered a mid-resource language (Gonzalez-Agirre et al., 2024). Out of the minority languages used in this research, the most comprehensive resources are available for Catalan. These include the AnCora corpus with over 500,000 tokens, which is also available on Universal Dependencies (Taulé et al., 2008). The AINA project[20] was established to use artificial intelligence to generate more content in Catalan and thus expand linguistic resources (Gonzalez-Agirre et al., 2024; Ramirez-Mitjans et al., 2024). The following datasets from Universal Dependencies were used for the experiments: ca_ancora-ud-train (447k tokens) for the example annotations and ca_ancora-ud-dev (60k tokens) for testing.

3.2.2 Galician

Galician has solid resources, although significantly less extensive than Catalan. There are several annotated corpora, including the CTG corpus used for this study, which contains over 100,000 tokens (Agerri et al., 2018). The largest corpus, CorpusNós, was developed specifically for training large language models, such as a 1.3B GPT-3 model and several small BERTa models (de-Dios-Flores et al., 2024). For the experiments, gl_ctg-ud-train (87k tokens) was used for the example annotations and gl_ctg-ud-dev (34k tokens) for testing. Despite efforts to provide more high-quality resources for Galician, it is considered a low-resource language (de-Dios-Flores et al., 2024).

3.2.3 Occitan

Occitan presents a particular challenge: there are currently no annotated datasets available for Occitan on the Universal Dependencies website. Occitan is a language with many variations, which poses challenges in the field of NLP for low-resource languages. In 2024, the CorpusArièja was introduced, containing around 42k tokens. It was annotated with POS tags, verbal inflection, and lemmatization, with the aim of training tagging tools and public NLP applications on it (Poujade et al., 2024). Since there is currently no data available in CoNLL-U format for the low-resource language Occitan, either on the Universal Dependencies website or from any alternative source, a gold standard was created for the experiments on Occitan.

[20] https://projecteaina.cat/

3.2.4 Datasets for the Comparison Languages

Next, we introduce the datasets for the comparison languages. The training sets were used to provide example annotations in the prompts, while the development sets were used as test data. For French, this included fr_gsd-ud-train (364k tokens) and fr_gsd-ud-dev (37k tokens). In the case of Spanish, sentences were drawn from es_gsd-ud-train (390k tokens) and es_gsd-ud-dev (38k tokens). For Portuguese, the data came from pt_gsd-ud-train (273k tokens) and pt_gsd-ud-dev (34k tokens). Finally, for Basque, the datasets eu_bdt-ud-train (73k tokens) and eu_bdt-ud-dev (24k tokens) were employed. In addition, a dataset with mixed example annotations was composed, containing annotations in French, Italian, Spanish, Romanian, and Portuguese. The annotations for French, Portuguese, and Spanish were taken from the test sets mentioned above, those for Romanian from ro_rrt-ud-train (185k), and those for Italian from it_vit-ud-train (242k).

3.2.5 Data Preparation

The datasets consisted of 50 sentences extracted from the development sets, available on the Universal Dependencies website, in the respective languages. These were then used as the gold standards against which the model performance was measured. To create the input files for the experiments, only the first three columns were retained, as well as the complete #sentiD and #text lines. The remaining columns were removed, as the task of the model was to complete them.

For the example annotations shown in the few-shot in-context learning prompts, sentences from the training sets were extracted from the Universal Dependencies website. Subsequently, files with ten example annotations per language were created, along with one additional 1-shot (long) and 1-shot (short) annotation, each in a separate file. For the mixed annotations, the following approach was used: For the 3-shot scenario, one sentence each in French, Spanish, and Portuguese was used. For the 5-shot scenario, one sentence each in French, Spanish, Portuguese, Italian, and Romanian was used, and for the 10-shot scenario, two sentences each of the five different languages were used. Since there is currently no suitable data available for Occitan, a gold standard was created for the experiments. The procedure for this task is described in the following chapter.

3.3 Creating a Gold Standard for Occitan

The following describes the creation of an appropriate gold standard for Occitan, structured as a treebank in CoNLL-U format. First, the data used for this purpose is presented, followed by the creation of the final dataset and the annotation and validation process.

3.3.1 Source Data

For the creation of the Occitan dataset with annotations in the form of a treebank in CoNLL-U format, sentences from the Restaure Corpus[21] (Bernhard et al., 2018) were selected. This corpus contains annotations with part-of-speech tags from three regional languages in France, namely Alsatian, Occitan, and Piccard. These were annotated as part of the Restaure Project, which aimed to provide computational resources and processing tools for the three regional languages of France. Linguists and NLP specialists collaborated on this project. Occitan is not a unitary language and consists of different varieties that can be divided into six major dialects. These dialects were not examined in detail in this work. As part of the Restaure project, the BaTelÒc text base (Bras & Vergez-Couret, 2016) was completed, which contains written texts from literature such as prose, drama, and poetry, as well as other genres such as technical texts and newspaper articles.

The dialectal diversity of the language is represented in the text corpus. In addition to selected texts from the BaTelÒc text corpus, texts from the Occitan newspaper Lo Jornalet[22] and a text from the virtual online library Ciel'dòc[23] were also selected. For the POS annotations, the texts were tagged with the POS tagger of the APERTIUM[24] translation platform. The first annotations were checked manually and then a supervised machine learning tagger, Talismane (Urieli, 2013), was trained on the corrected corpus. The rest of the corpus was annotated by this tagger, whose output was also corrected manually (Bernhard et al., 2018). The token ID, word, lemma, and POS annotations could be used to create the gold standard for this research. The POS tags were corrected in rare cases. In addition, the Restaure corpus contained further information, namely morphological features, but these were not taken into account in the subsequent annotation process. The process of creating the dataset, as well as its annotation and validation, described below.

[21] http://restaure.unistra.fr/

[22] https://www.jornalet.com/

[23] http://www.cieldoc.com/

[24] https://www.apertium.org/

3.3.2 Annotation and Validation Process

The first step was to compile all the text data from the Restaure corpus for Occitan. Then, using the Python module `random`[25], 60 sentences were selected, 50 for the creation of a gold file for the test dataset and ten for the example annotations. Since the text data of the Restaure corpus is available as a CSV file, these 60 sentences were then converted to CoNLL-U format, and sentence ID (#sentID) and text line (#text) were added. Similar to the approach taken by Bernhard et al. (2018), the text was first parsed by a parser and then manually revised. The already tokenized text was parsed with the Stanza parser[26]. On the one hand, with the Stanza parser in the Catalan language, and on the other hand, in the French language. These languages were chosen because of their linguistic proximity to Occitan.

Unfortunately, parsing with the Stanza parser did not deliver the desired output, as the Catalan parser made many errors in the annotations and the French parser often skipped lines and annotated the head with an X. Therefore, even more attention was paid to accuracy in the next step. During manual revision, individual words were translated using the Glosbe Translator[27], and the POS tags for the syntactic function of the tokens were taken into account. The focus was primarily on the head annotations and the DEPREL annotations, so that in the end, TokenID, Word, Lemma, UPOS, Head, and DEPREL are available according to Universal Dependencies standards, strictly following the UD guidelines on the Universal Dependencies website.

After manual revision, the treebank was validated using the `validate.py`[28] file. The `validate.py` file inspects the content in a CoNLL-U file and detects format errors, spelling mistakes, syntax errors, etc. The errors detected by `validate.py` were corrected manually until all errors had been eliminated. In some cases, non-syntactic information (MISC) was also added. The annotated dataset was then complete and ready to be used for the experiments.

[25] https://docs.python.org/3/library/random.html

[26] https://stanfordnlp.github.io/stanza/

[27] https://translate.glosbe.com/oc-en

[28] https://github.com/UniversalDependencies/tools/blob/master/validate.py

4. Experiments

This section describes the detailed implementation of the experiments. The first step was prompt engineering, which aimed to find the most suitable prompt for the subsequent zero-, one-, and few-shot in-context learning experiments. Different text formulations and prompt structures were tested in this process. The following sections explain the in-context learning experiments. First, the settings in which the experiments were conducted are explained. This refers to which language pairs (i.e., combinations of target and source languages) were tested in which shot scenario. Here, the term target language refers to the language of the input file to be annotated by the model, whereas the source language denotes the language of the example annotations. This is followed by a description of the implementation of the experiments in an unconstrained setting, in which the output was freely generated using the model's API, to be used as a benchmark for the constrained decoding experiments. Finally, the constrained decoding experiments are outlined. The results of the in-context learning experiments are presented in chapter 5. The entire code for reproducibility of the results and the data is available on GitHub[29].

4.1 Prompt Engineering

An important prerequisite for experiments with generative language models is well-considered prompt design. Effective prompt engineering is crucial for large language models to reach their full potential. But what is a prompt?

"A prompt is a text-based input that is fed to a language model to guide its output." (Marvin et al., 2024, p. 389)

A prompt is used to give instructions to a language model and provide context for achieving a desired task. A prompt usually contains instructions and can also provide input data, context, and output indicators (Marvin et al., 2024). Prompt engineering describes a process of designing and optimizing these prompts to generate the desired output. White et al. (2023) describe prompt engineering as an increasingly important skill set required to conduct a successful conversation with an LLM. Writing prompts is already a form of programming. Effective prompts not only

[29] https://github.com/maralb99/syntax-parsing-icl-romance-langs.git

lead to better output by resulting in greater accuracy and more useful output but can also save time and resources (Marvin et al., 2024).

According to Marvin et al. (2024), there are five steps that can help you create an effective prompt:

1. Define the goal: First, provide clear instructions on what the goal of the prompt is.
2. Understand the capabilities of the model: To be able to use the full potential of a language model, it is important to understand both the capabilities and limitations of the model. This also includes clarity about what types of responses a model can generate (text, audio, images, etc.).
3. Choose the right prompt format: Choosing a clear and precise prompt format can also improve the model output significantly. Prompt formats in the form of templates can also be helpful, presenting placeholders for specific elements such as training or test examples and showing the structure of the desired output (Zhao et al., 2021). White et al. (2023) created a catalog presenting prompt engineering techniques in the form of prompt patterns designed to solve common problems that arise in conversations with LMs. The use of prompt patterns increased the model's ability to solve complex tasks. It was also found that prompt patterns can be generalized to a multitude of domains.
4. Provide context: The role of context provided to an LLM was often underestimated in the past, but it is essential for better understanding the prompt and providing more precise information. Without clear context, the model often provides overly general responses. Nevertheless, providing context can promote targeted content generation and produce responses in the desired format (Zhao et al., 2021).
5. Test and revise the prompt: It is fundamental to test and evaluate the prompt (Trautmann et al., 2022). Subsequently, the prompt can be refined further so that the output of the language model is continuously optimized. Even minor modifications can significantly alter the predictions of the language model (Arora et al., 2022).

Following these steps, the best prompt for the upcoming in-context learning experiments was sought. In the prompt engineering process, the prompts were tested on a French development set and a Spanish development set. For this purpose, 50 sentences were extracted from fr_gsd-ud-dev.conllu and es_gsd-ud-dev.conllu. The tokenization of these development sets was adapted, which means that the first three columns, namely the token ID, word, and lemma, formed the input for these experiments. The other columns, which are common for the CoNLL-U format, were to be generated by the model. Eight different prompts were tested. The development of the prompts was carried out in an exploratory and iterative process. The process began with a basic

prompt, which was successively expanded with additional information to optimize performance. Each prompt created was systematically evaluated using the `eval.py`[30] file from Universal Dependencies by testing it on both the Spanish development set and the French development set. The precision scores from UPOS, UAS, and LAS were used as comparison metrics. To determine the optimal performance of the model, all prompts were also tested with two different hyperparameter configurations. Once, the hyperparameter temperature (t) was set to $t_1 = 0.0$ and once to $t_2 = 0.3$. The temperature hyperparameter is an important setting in stochastic processes, responsible for the randomness of the generation process and regulating the balance between creativity, i.e., more randomness, and precision, i.e., less randomness, in a generation process. In text generation, higher temperatures lead to more unpredictable texts that are also less coherent, while lower temperatures lead to higher-quality texts, but with less variation (Peeperkorn et al., 2024; S. Zhang et al., 2024).

It started with a basic prompt, which was gradually enriched with increasing amounts of information. Table 1 presents the eight prompts that were tested here.

Prompt_ID	Prompt
Prompt 1	"Annotate" f"{tokenized_input}"
Prompt 2	"Please annotate the following sentence using the Universal Dependencies standard" f"{tokenized_input}"
Prompt 3	"Please annotate the following sentence using the Universal Dependencies standard, ensuring the output adheres to the CoNLL-U format strictly (10 fields, tab-separated, with ID starting each line). The input is already tokenized with the columns ID, FORM, and LEMMA provided. Complete the remaining columns (UPOS, XPOS, FEATS, HEAD, DEPREL, DEPS, MISC)." f"{tokenized_input}"
Prompt 4	"Please correctly annotate the following sentences with POS Tags and Dependency Relations using the Universal Dependencies standard. Ensure that the output strictly adheres to the CoNLL-U format (10 fields, tab-separated, starting with ID in each line). The input already includes the columns ID, FORM, and LEMMA. Complete the remaining columns (UPOS, XPOS, FEATS, HEAD, DEPREL, DEPS, MISC) accordingly." f"{tokenized_input}"

[30] https://github.com/UniversalDependencies/tools/blob/master/eval.py

Prompt 5	"Please annotate the following sentence using the Universal Dependencies standard, ensuring the output adheres to the CoNLL-U format strictly (10 fields, tab-separated, with ID starting each line). The input is already tokenized with the columns ID, FORM, and LEMMA provided. Complete the remaining columns (UPOS, XPOS, FEATS, HEAD, DEPREL, DEPS, MISC). Ensure that the root word has a HEAD value of 0, and there is exactly one root in each sentence. The HEAD value of other tokens must point to their correct governor. Return only the output in CoNLL-U format, without any explanations or additional text:\n\n" f"{tokenized_input}"
Prompt 6	"Annotate the tokenized input in Universal Dependencies (UD) standard. Input contains pre-filled columns (ID, FORM, LEMMA). Complete the remaining columns (UPOS, XPOS, FEATS, HEAD, DEPREL, DEPS, MISC) strictly in CoNLL-U format (10 fields, tab-separated, one token per line). Ensure: • One HEAD value is 0 (the root token). • Other tokens have HEAD pointing to the correct governor. • Maintain syntactic and semantic correctness according to UD guidelines. Provide only the CoNLL-U output, no additional explanations or comments." f"{tokenized_input}"
Prompt 7	"Annotate the tokenized input in Universal Dependencies (UD) standard. Input contains pre-filled columns (ID, FORM, LEMMA). Complete the remaining columns (UPOS, XPOS, FEATS, HEAD, DEPREL, DEPS, MISC) strictly in CoNLL-U format (10 fields, tab-separated, one token per line). Ensure: One HEAD value has to be 0 (the root token). Multi ID tokens (e.g. 18-19) must be considered but should not be annotated. Provide only the CoNLL-U output, no additional explanations or comments." f"{tokenized_input}"
Prompt 8	"Please annotate the following sentence using the Universal Dependencies standard, ensuring the output adheres to the CoNLL-U format strictly (10 fields, tab-separated, with ID starting each line). The input is already tokenized with the columns ID, FORM, and LEMMA provided. Complete the remaining columns (UPOS, XPOS, FEATS, HEAD, DEPREL, DEPS, MISC). Multi-ID-tokens (like 18-19) should be considered but not be annotated." f"{tokenized_input}"

Table 1: list of prompts tested in the prompt engineering process

As can be observed, additional information intended to improve performance was gradually added to the prompts. The results were unexpected in some cases, as more information in the prompts did not automatically lead to higher performance values. The following two graphs show the performance metrics on the French development set and the Spanish development set. The x-axis displays the different prompt-temperature pairings. The language codes, es or fr, indicate the language on which the prompt was tested. This is followed by a number, being the prompt ID presented in the previous table. Finally, the hyperparameter temperature follows, set to either 0.0 or 0.3. Fr-1-0.0 thus shows the precision scores for the output of prompt 1 with the temperature setting $t_1 = 0.0$, etc.

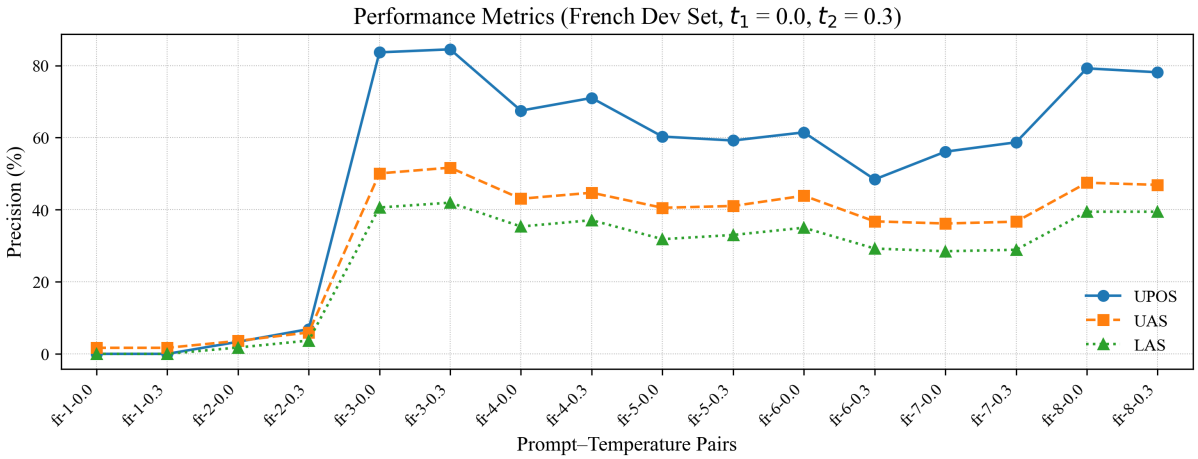


Figure 1: performance scores for French for each prompt-temperature pair

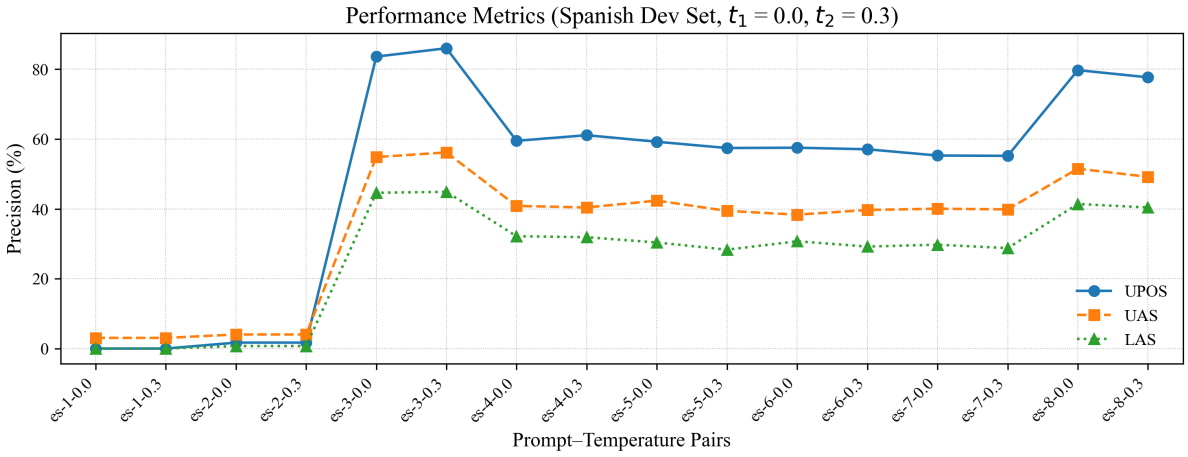


Figure 2: performance scores for Spanish for each prompt-temperature pair

As can be seen in the graphs, the precision scores of the output of the prompts are not always identical for the French and Spanish dev sets, but the performance curves are very similar. The highest quality output was achieved using prompt 3 with the temperature setting at $t_2 = 0.3$

on both the Spanish and French development sets. Consequently, this prompt was employed in the subsequent experiments for the zero-shot, one-shot, and few-shot scenarios. Although subsequent prompts attempted to optimize the model's ability to annotate the given input text with Universal Dependencies, this could not be achieved. In general, the model has difficulty recognizing and annotating multi-ID tokens used for contractions, such as *dels* (*de* + *les*) in Catalan. In addition, the model often sets multiple roots, no roots at all or annotates heads that exceed the number of tokens in a sentence. The prompts were formulated to reduce these errors. Nevertheless, after prompt 3, this was no longer achieved: only prompt 8 delivered comparable performance values.

As expected, the output generated by prompt 1 was not in CoNLL-U format and therefore could not be evaluated with the `eval.py` file, which is why the placeholder file was used for evaluation. Prompt 2 also yielded very low precision scores (<6%) for UPOS, UAS, and LAS. As already mentioned, prompt 3 achieved the highest UPOS, UAS, and LAS scores in both languages. The results for prompts 4-7 were again considerably lower. To investigate whether the difference in output between prompt 3 and prompt 4 is statistically significant, a paired t-test (t-test for dependent samples) was performed with Python to calculate the p-value. The paired t-test was chosen because the values are dependent on each other in pairs, as they are different prompts tested on the same dataset. This statistical analysis shows that prompt 3 performs significantly better than prompt 4 in the metrics UAS ($p = 0.003$) and LAS ($p = 0.026$), assuming a significance level of $\alpha = 0.05$. The difference in performance for the UPOS metric ($p = 0.057$) is just above the threshold of statistical significance (Bortz & Schuster, 2011; Fahrmeir et al., 2023).

Despite the process's design for prompt optimization, the graphs show a decline in performance results from prompt 4 to prompt 7. It is also interesting to note that the hyperparameter setting temperature (t) contributed differently to performance from prompt to prompt. Sometimes the same prompt with the temperature setting $t_1 = 0.0$ achieved higher values, other times the prompt with the temperature setting $t_2 = 0.3$ did. The prompts explicitly attempted to counteract the errors that the model tends to make, such as misannotations in multi-ID tokens or root errors. In this case, however, the errors could not be corrected by prompt engineering. This chapter showed the process of selecting the prompt used for the in-context learning experiments. The next section explains the structure of the experiments.

4.2 Experiment Settings

This chapter explains the different experimental settings used in this research. All experiments were conducted in Python[31] (version 3.12.2). The unconstrained decoding experiments were conducted in Google Colab[32], while the constrained decoding experiments were conducted via the Slurm cluster (Yoo et al., 2003) access of the University of Vienna. The creation of the input datasets has already been explained. The first experimental setting is free generation without constraints, the results of which are to serve as a baseline. The second experimental setting is constrained decoding with structural constraints, and the third is constrained decoding with both structural and linguistic constraints. Table 2 shows the settings in which free generation was performed without constraints:

Target lang	0-shot	1-shot (short and long), 3-, 5- and 10-shot	3-,5- and 10-shot
ca	✓	ca, es, oc	mixed annotations
gl	✓	gl, es, pt	mixed annotations
oc	✓	oc, fr, ca	mixed annotations
fr	✓	-	-
es	✓	-	-
pt	✓	-	-
eu	✓	-	-

Table 2: unconstrained generation experiments: target languages, shot settings and corresponding source languages

The table provides an overview of the scenarios, which target languages were used in which shot settings, and which source languages were used in the unconstrained setting. For all languages – Catalan (ca), Galician (gl), Occitan (oc), French (fr), Spanish (es), Portuguese (pt), and Basque (eu) – 0-shot experiments were first conducted in which no examples were shown. In addition, 1-shot experiments with unconstrained decoding were also conducted for Catalan, Galician, and Occitan, in which one short and one long example annotation were shown. The aim was to investigate whether the length of the annotation affects performance. In addition, 3-, 5-, and 10-shot experiments were also conducted for these three languages. The table specifies the source languages employed in the different shot settings: for Catalan, the sources were Catalan, Spanish, and Occitan; for Galician, Galician, Spanish, and Portuguese; and for Occitan,

[31] https://www.python.org/

[32] https://colab.research.google.com

Occitan, French, and Catalan. Mixed annotations were also used in the few-shot scenarios, but only in the 3-, 5-, and 10-shot settings.

In contrast, French, Spanish, Portuguese, and Basque were only used in the 0-shot scenario, and no further examples were provided. Following the experiments in free generation, experiments were conducted in which only structural constraints were applied: However, these were only carried out in the 0-shot scenario for each target language.

This was followed by experiments in the decoding form, which combines structural and linguistic constraints. In this form of decoding (constrained decoding), 1-shot and few-shot experiments were also conducted for French, Spanish, Portuguese, and Basque as target languages—in addition to Catalan, Galician, and Occitan. This allows for a direct comparison between high-resource Romance languages and a non-Romance language such as Basque in the context of few-shot constrained decoding. This allows the results for minority languages to be classified more clearly. Table 3 provides an overview of the target languages, the shot settings used, and the source languages used in each scenario in the constrained decoding setting.

Target lang	0-shot	1-shot (short, standard and long), 3-, 5- and 10-shot	3-,5- and 10-shot
ca	✓	ca, es, oc	mixed annotations
gl	✓	gl, es, pt	mixed annotations
oc	✓	oc, fr, ca	mixed annotations
fr	✓	fr, es, oc, ca	mixed annotations
es	✓	es, fr, ca	mixed annotations
pt	✓	pt, es, fr, gl	mixed annotations
eu	✓	eu, oc, fr, es	mixed annotations

Table 3: constrained decoding with structural and linguistic constraints: target languages, shot settings and corresponding source languages

Zero-shot experiments were conducted for all target languages. In addition, one-shot experiments (with short, standard, and long examples) and few-shot experiments with 3, 5, and 10 examples were used. In the constrained setting, the same shot settings and source languages were applied to the Romance minority languages as in the unconstrained decoding. The high-resource Romance languages and Basque were paired with the following source languages: French with French, Spanish, Occitan, and Catalan; Spanish with Spanish, French, and Catalan; Portuguese with Portuguese, Spanish, French, and Galician; and Basque with Basque, Occitan,

French, and Spanish. In all these cases, mixed annotations were also tested in the few-shot scenarios 3-shot, 5-shot, and 10-shot.

For a better overview, the combination of target and source languages is referred to below as language pairs, with the target language always mentioned first. For example, the pair *ca_oc* refers to Catalan as the target language and Occitan as the source language. Having described the experimental settings in this section, the exact steps taken in the experiments will now be described.

4.3 Unconstrained Decoding

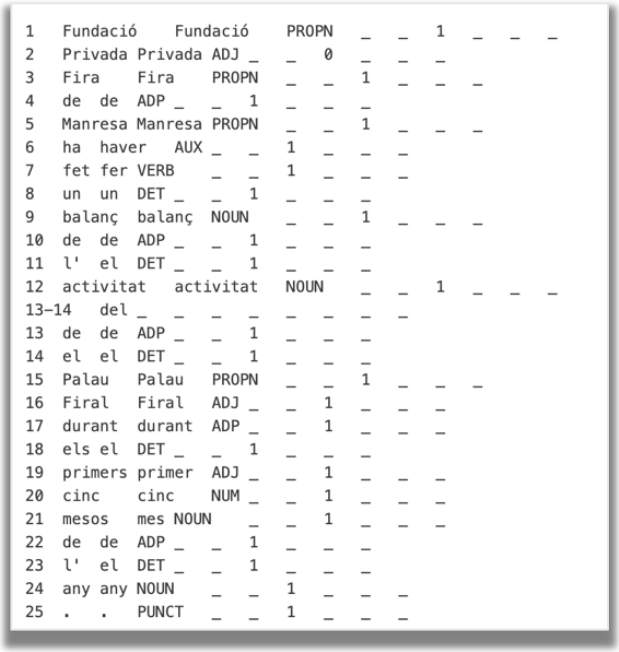
To create comparative values for the parsing results of the constrained decoding parser, experiments were first conducted in an unconstrained setting. In this step, free generation was performed using the API of the Aya Expanse 32B model. Prompt 3 was used as the text prompt and the temperature was set to 0.3. The input file is a CoNLL-U file in which the first three columns are given (token ID, word, lemma). Next, a prompt is created that integrates the text prompt, takes one sentence from the input file, and asks the model to annotate the remaining 7 columns. In the few-shot scenarios, the example annotations were also integrated, which were displayed in a complete CoNLL-U format with all 10 columns.

```
def generate_conll_u_annotation(tokenized_input, example_annotation=""):
    prompt = (
        "Please annotate the following sentence using the Universal Dependencies standard, "
        "ensuring the output adheres to the CoNLL-U format strictly (10 fields,tab-separated, with ID starting each line). "
        "The input is already tokenized, with the ID, FORM, and LEMMA columns provided. "
        "Complete the remaining columns (UPOS, XPOS, FEATS, HEAD, DEPREL, DEPS, MISC).\n\n"
        f"Here is an example of the expected format:\n{example_annotation}\n\n"
        "Sentence to annotate:\n"
        f"{tokenized_input}"
    )
```

Figure 3: prompt structure

In the unconstrained decoding experiments, no constraints were enforced during generation, which means that the model can produce invalid structures and invalid dependency trees. These steps led to the generation of a raw output file containing the unmodified output. Although this file contained the desired annotations, it was highly unstructured and also contained additional comments from the language model. But to be able to evaluate the annotations with the `eval.py` file, the output had to follow a strict CoNLL-U format. Therefore, after the free generation, an extra processing step with post-processing constraints was done. This step included aligning the fields and adding missing tokens. In addition, it was important to ensure that the format structure of a CoNLL-U file was retained, with separate tabs and no additional comments from the model. Nevertheless, this step did not include validating the dependency tree structure, i.e., checking for cycles or ensuring that there is a single root. There was no

verification of the HEAD pointing to a valid token or of valid POS tags or dependency labels being assigned. The output, i.e., the system file for evaluation, was already in CoNLL-U format, but it contained several errors that resulted in the evaluation being unable to be performed. In the next step, the records were therefore checked, revealing errors in the number of columns, errors in the HEAD values showing non-existent tokens, multiple roots and no roots errors, as well as invalid dependencies. After the system file had been analyzed and the errors had been identified, placeholders were inserted in this format.



```
1   Fundació    Fundació    PROPN   _   _   1   _   _   _
2   Privada Privada ADJ _   _   0   _   _   _
3   Fira    Fira    PROPN   _   _   1   _   _   _
4   de  de  ADP _   _   1   _   _   _
5   Manresa Manresa PROPN   _   _   1   _   _   _
6   ha  haver   AUX _   _   1   _   _   _
7   fet fer VERB    _   _   1   _   _   _
8   un  un  DET _   _   1   _   _   _
9   balanç  balanç  NOUN    _   _   1   _   _   _
10  de  de  ADP _   _   1   _   _   _
11  l'  el  DET _   _   1   _   _   _
12  activitat   activitat   NOUN    _   _   1   _   _   _
13-14   del _   _   _   _   _   _   _
13  de  de  ADP _   _   1   _   _   _
14  el  el  DET _   _   1   _   _   _
15  Palau   Palau   PROPN   _   _   1   _   _   _
16  Firal   Firal   ADJ _   _   1   _   _   _
17  durant  durant  ADP _   _   1   _   _   _
18  els el  DET _   _   1   _   _   _
19  primers primer  ADJ _   _   1   _   _   _
20  cinc    cinc    NUM _   _   1   _   _   _
21  mesos   mes NOUN    _   _   1   _   _   _
22  de  de  ADP _   _   1   _   _   _
23  l'  el  DET _   _   1   _   _   _
24  any any NOUN    _   _   1   _   _   _
25  .   .   PUNCT   _   _   1   _   _   _
```

Figure 4: placeholder example for Catalan

If the error only affected one token (i.e., one line), e.g., a "`Token has 4 columns instead of 10 in sentence 36`" error, this line was replaced with a placeholder line. If an error affected an entire sentence, such as "`Multiple roots found`," the entire sentence was replaced with placeholders. To automate these processes, avoid having to switch manually between different files, and generate an automated workflow, the Python module `subprocess`[33] was used. Replacing the erroneous lines creates a corrected system file, which can then be used for evaluation with the `eval.py` file. The results of the unconstrained decoding experiments are described and analyzed in more detail in chapter 5. As can be seen, free generation without constraints entails some inconveniences, such as the need for post-processing steps so that the output can be evaluated according to Universal Dependencies standards. The

[33] https://docs.python.org/3/library/subprocess.html

constraints that were set to increase performance and obtain a more structured output are explained in the next section.

4.4 Constrained Decoding

Constrained decoding is a technique for enforcing constraints on the output of a language model and provides the ability to control the output of text generation without retraining and without changing the model architecture. Constrained decoding was limited to open-source models that allowed access to their next-token logits during generation (Geng et al., 2024). This limitation was overcome by Geng et al. (2024) with sketch-guided constrained decoding (SketchGCD), which also works with black-box LLMs. Since the Aya Expanse 32B model is an open access model (J. Dang, Singh, et al., 2024), SketchGCD is not used here, as the internal logits can be accessed. Constrained text generation allows users to generate texts that must meet certain requirements. This approach is particularly useful when precisely controlled output is required. Constrained text generation can improve the performance of many downstream tasks, such as machine translation, dialog response generation and recipe generation, as well as the conversion of tables to continuous text (Chen & Wan, 2024). Chen and Wan (2024) distinguish between three categories of constraints in their article:

- Lexical Constraints: A set of keywords is given, which can be based on grammatical rules, semantic restrictions or stylistic guidelines. The model is forced to integrate these keywords into the output text.
- Structural Constraints: This includes those restrictions that relate to the structure of the output, such as the number of sentences, words, tokens, etc. and other factors that affect the structure.
- Relation Constraints: This refers to structural constraints that require certain relationships between lexical elements in the generated sentence, e.g. syntactic dependencies, semantic roles or other domain-specific relationships (Bastan et al., 2023).

The use of constrained decoding is highly useful in syntactic parsing with a generative language model. As already demonstrated in the last chapter, the free, unconstrained generation with the language model is less structured: even if the model recognizes the correct syntactic relations in the input sentences, the results are frequently not provided in an evaluable CoNLL-U format. Explanations are often provided in addition to the annotation of the sentences. Furthermore, as already mentioned, there often is an incorrect annotation of multi-ID tokens and thus an incorrect number of tokens to be annotated, which in turn leads to errors in the token IDs, which can then

cause difficulties during evaluation. Since the free generation was often too unstructured for the evaluation, which requires a strict CoNLL-U format, constrained decoding can help to ensure that the output receives the structure that is necessary for the evaluation. The following section explains how the setup for these experiments was constructed and which forms of constrained decoding were used here.

4.4.1 Structured Prompting

The script by Blevins et al. (2023) for sequence labeling tasks, which utilizes constrained decoding with LLMs and in-context learning, served as the basis for this script. POS tagging was mainly used here, but NER and sentence chunking were also applied. The approach is based on three principles: sequential left-to-right processing, constrained generation through logit masking, and few-shot in-context learning with template-based demonstrations. The system processes each token independently, predicts individual labels, and takes into account the context of previously processed tokens and their predicted labels. To perform dependency parsing, this approach required some adaptations.

4.4.2 Structural Constraints

The script was developed on a development set subset with 10 sentences in French. For an easier evaluation with the `eval.py` script of Universal Dependencies, it was decided to specify the input as well as the in-context examples and the output in CoNLL-U format. As a simple approach is less suitable for recognizing dependencies, a two-phase sequential approach was implemented. In the first pass, all POS tags are predicted for all tokens. In the second pass, the heads and the dependency relations are predicted, but the POS information is also taken into account as context. In this step, structural constraints were also introduced to achieve a valid tree structure. These included the identification of a single root, cycle detection and breaking as well as connectivity validation, which means that all tokens should be reachable via the root. The `find_best_root()` function implements a single heuristic in which verbs are preferred as roots. The `fix_dependency_tree()` function repairs structural violations with systematic tree traversal. In addition, the parser is forced to select only from a list of allowed dependency labels. Now the tree structure is mathematically correct and can be validated and evaluated. But while the POS scores are already very high (around 90% or higher), the labeled attachment scores and unlabeled attachment scores are low: The UAS is usually between 22% and 35%, and LAS between 16% and 26%. Despite leading to validation and evaluability, structural constraints still

violate fundamental syntactic principles. For this reason, it was decided to develop the script even further and to include linguistic principles.

4.4.3 Structural and Linguistic Constraints

At the last stage of script development, linguistic constraints were used to encode universal syntactic dependencies and language-specific patterns. According to Chen and Wan (2024), these linguistic constraints could also be categorized as relation constraints. This change addresses the performance limitations of the structural-constraints-only parser for UAS and LAS. It is also important to mention that the structural constraints from the last stage (4.4.2) are retained. The linguistic constraints are derived from the information on syntactic dependencies on the Universal Dependencies website. The basic innovation here is that not every token can attach to another token; instead, the parser uses the part-of-speech information to restrict the head candidates to linguistically plausible options. To improve parsing results, linguistic heuristics were developed that are specifically adapted to the French language. Their design considers the basic principles of dependency parsing and the dependency formalisms according to Jurafsky and Martin (2025). In addition, the heuristics for annotating POS tags and dependency relations are based on the guidelines provided by Universal Dependencies (Nivre, 2020; Nivre et al., 2016).

Another theoretical basis was the mean dependency distance (MDD). H. Liu (2008) demonstrated that the mean dependency distance in natural texts of multiple languages typically lies between 1.8 and 3.7 tokens. This indicates that dependency distances in human languages are systematically constrained to a narrow range (H. Liu, 2008). The specific parameter values were then motivated by language-specific considerations for French. Since this is a high-resource language, the assumption is also made that positive effects from cross-lingual transfer could also be transferred to other Romance languages. The following heuristics were implemented in the script:

- determiners attach to nearby nouns (within 3-4 tokens), assuming determiners precede their head noun
- auxiliaries attach to main verbs
- adjectives attach to nearby nouns (±3 tokens)
- adverbs attach to verbs, adjectives, or other adverbs
- punctuation attaches to nearby content words (within 5 tokens)

For determiners, a range of 3-4 tokens from the noun was defined. This decision was motivated by the possibility of intervening adjective sequences between determiners and nouns, e.g. *une belle grande maison* (English: *a beautiful large house*). In the case of adjectives, the heuristic specifies a symmetric placement, allowing for their occurrence both before and after the noun. This reflects the structural flexibility of French, where adjectives may precede the noun (e.g., *une belle maison* – English: *a beautiful house*) or follow it (e.g., *une maison rouge* – English: *a red house*). The remaining heuristics are also based on typical regularities in French grammar, such as the predominant assumption of an SVO word order (Grevisse & Goosse, 2016).

All token distances defined in the script were systematically evaluated through an iterative process, and all the distance values that produced the highest performance results were retained. For instance, the distance for punctuation was extended from the original 2 tokens, as it was found that punctuation marks in French are more flexible in terms of their position, particularly in contexts such as abbreviations, allowing for a larger number of intervening tokens between the punctuation mark and associated token.

The parser considers the POS tag of the dependent, the POS tag of the head and the relative position to determine the dependency relation. Examples for this approach are shown in table 4:

Scenario	Dependent POS	Head POS	Position	DEPREL
Noun before Verb	NOUN/PROPN/PRON	VERB/AUX	before head	subj nsubj:pass csubj
Noun after Verb	NOUN/PROPN/PRON	VERB/AUX	after head	obj iobj obl
Determiner to Noun	DET	NOUN/PROPN	any	det
Adjective to Noun	ADJ	NOUN/PROPN	any	amod
Adverb to Verb	ADV	VERB/AUX	any	advmod
Preposition Relations	ADP	NOUN/VERB	any	case mark

Table 4: linguistic heuristics used for constrained decoding

In addition, the parser employs a sophisticated hierarchy to select the linguistically most appropriate root, which follows the scheme VERB > AUX > NOUN/PROPN > others. The parser

calculates the joint probabilities for head-dependency pairs and then selects the combination that is most probable. Special attention is given to the correct processing of multiword tokens, which are often used in contractions. Furthermore, care is taken to ensure correct output in CoNLL-U format.

5. Results

This chapter is devoted to the results of the 0-, 1-, and few-shot in-context learning experiments, the implementation of which was explained in Chapter 4. In Chapter 6.1.2, these values are analyzed again in the framework of a quantitative analysis and evaluated and classified for statistical significance.

5.1 Results for the 0-shot Experiments

Table 5 presents the results of the 0-shot scenario for the unconstrained decoding experiments and the constrained decoding experiments, once with structural constraints only and once with structural and linguistic constraints. Due to the low results for UAS and LAS, the 0-shot setting is the only scenario that was performed with constrained decoding using structural constraints only. The best results for each target language are highlighted in boldface.

Target	Unconstrained			Structural Constraints			Struct. + Ling. Constraints		
lang	UPOS	UAS	LAS	UPOS	UAS	LAS	UPOS	UAS	LAS
fr	84.51	**51.68**	41.96	**95.73**	30.74	25.13	**95.73**	48.91	**43.13**
es	86.02	**56.18**	**44.89**	**92.56**	27.69	20.88	**92.56**	43.73	36.29
pt	76.98	39.98	30.94	**91.52**	31.99	23.42	**91.52**	**49.03**	**41.44**
eu	68.90	30.24	17.01	**76.29**	34.54	**20.62**	**76.29**	**35.05**	15.46
ca	69.21	28.39	21.86	**92.36**	22.70	17.12	**92.36**	**44.34**	**36.31**
gl	82.44	33.15	23.92	**91.22**	23.64	16.38	**91.22**	**44.79**	**38.55**
oc	74.64	35.32	24.88	**85.47**	29.62	19.59	**85.47**	**44.48**	**33.67**

Table 5: 0-shot performance scores for unconstrained and constrained settings; scores in %

Looking at this table, it is evident that high performance scores were already achieved in the unconstrained setting for the target languages French and Spanish. However, the situation is different for the Romance minority languages and Basque. For Galician, the UPOS score is quite high at 82.44%, but the other UPOS values, which range between 68.90% and 76.98%,

are noticeably lower. The UAS values for the lower-resource languages range between 28.39% and 35.32%, while the LAS values range between 17.01% and 24.88%. The values for the target language Portuguese are clearly lower than for French and Spanish, but higher than those for the lower-resource languages. Looking at decoding with structural constraints only, the table shows that the UPOS values have been considerably increased and are above 90% for all target languages except Occitan and Basque. Nevertheless, when looking at the UAS scores and LAS scores in these experiments, a sharp drop in performance can be observed compared to the unconstrained decoding. This leads to the conclusion that applying structural constraints alone improves a model's POS tagging ability but is less suitable for syntactic parsing. To improve the model's syntactic parsing ability, linguistic constraints based on the French language were added to the structural constraints. The UPOS values did not change with the addition of linguistic constraints, which underscores that UPOS values can be optimized by structural constraints alone. Even though the UAS and LAS values for the target language Spanish experienced a considerable loss compared to the unconstrained settings, and the UAS score for French is also below that of the unconstrained settings, substantial gains were recorded for the other target language values, especially for the Romance minority languages and Portuguese. Here, the UAS values per language were improved by around 10%, and for Catalan by as much as 16%. For minority languages as target languages, the improvement in LAS is around 9% for Occitan and around 15% for Galician and Catalan. For Basque, the change led to a lower LAS score. This could be because Basque is not a Romance language and therefore does not benefit from linguistic constraints that have been adapted to Romance languages.

5.2 Results for Minority Languages

After presenting the results obtained in the 0-shot scenario in the previous section, this chapter focuses on the results of the in-context learning experiments for the Romance minority languages as target languages. The tables with the complete results can be found in the appendix. The aim is to compare the results of the unconstrained decoding experiments with those of the constrained decoding experiments and to determine whether the model performance improves with increasing shot numbers. Since the experiments with purely structural constraints were conducted exclusively for the 0-shot scenario, from this point on, constrained decoding will always refer to decoding with both structural and linguistic constraints. Table 6 shows the model's performance for Catalan as the target language under different few-shot settings. The rows distinguish between language pairs (ca_ca, ca_es, ca_oc, ca_mixed) and the number of examples shown (#shots). The columns under "Unconstrained Decoding" and "Constrained

Decoding" show the accuracy of the UPOS tags, the unlabeled attachment scores (UAS), and the labeled attachment scores (LAS) in percent for the two decoding settings. Results for 0-shot, 1-shot with long annotations, 3-shot (only for mixed annotations), and 10-shot are displayed separately. The best scores in the tables are shown in bold.

lang_pair	#shots	Unconstrained Decoding			Constrained Decoding		
		UPOS	UAS	LAS	UPOS	UAS	LAS
ca	0	69.21	28.39	21.86	92.36	44.34	36.31
ca_ca	1 (long)	77.58	40.10	32.18	93.70	46.12	39.43
	10	69.10	37.42	31.29	**95.59**	45.51	39.04
ca_es	1 (long)	84.38	**52.09**	**41.61**	92.69	44.73	38.43
	10	76.63	45.23	37.70	92.58	43.39	37.48
ca_oc	1 (long)	83.32	47.74	36.87	92.53	44.56	38.04
	10	86.73	48.35	39.88	92.25	45.45	39.10
ca_mixed	3	72.39	39.21	30.23	91.91	45.06	38.32
	10	84.77	45.79	35.86	92.81	46.18	39.60

Table 6: performance for Catalan target language in 1-shot (long) and 10-shot settings; 3-shot results shown only for mixed annotations; scores in %

The first thing that stands out is that the unconstrained decoding results show much greater variation in the UPOS values than in the constrained decoding. It is also noteworthy that showing an example, whether short or long, in the unconstrained settings already led to much higher performance values than showing no examples at all. Even though the performance values of 0-shot and 1-shot (long) could also be increased in the constrained setting, this increase is far less substantial. A surprising finding is that, contrary to expectations, higher shot numbers do not automatically lead to higher performance values. Furthermore, there is no discernible pattern as to when a higher shot number leads to higher performance values. This also varies between the different language pairs in unconstrained and constrained settings. For example, in the ca_ca language pair in the unconstrained setting, there was a clear performance increase from 0-shot to 1-shot, but then a drop in the 3-shot setting. In the 10-shot setting, the UPOS value achieved was again higher than in the 3-shot setting, but lower than in the 0- and 1-shot settings. Meanwhile, experiments in the constrained setting of the same language pair yielded different results: There was also a notable improvement from 0-shot to 1-shot (long), but when comparing 1-shot (long) and 10-shot UAS and LAS, the 10-shot scenario is lower. The UPOS of the 10-shot scenario (95.59%) is the highest UPOS score for this language pair, as is also the

case for this target language. For the language pair ca_mixed in the constrained setting, the highest UPOS, LAS, and UAS values were achieved in the 10-shot scenario, indicating that results could be improved with increasing shot scenarios. The only exceptions are a higher UAS in the 3-shot (45.06%) than in the 5-shot (44.95%) scenario, and a higher UPOS in the 0-shot (92.36%) than in the 3-shot (91.91%) scenario. Nevertheless, this pattern does not apply to the same language pair in the unconstrained setting. The following section now continues with the analysis of the performance values for the target language Galician in order to identify possible parallels and differences. Table 7 shows the performance values Galician as target language under different shot settings. The rows distinguish between language pairs (gl_gl, gl_es, gl_pt, gl_mixed) and the number of examples shown (#shots). The columns "Unconstrained Decoding" and "Constrained Decoding" indicate the accuracy of the UPOS tags as well as the UAS and the LAS in percent for the two decoding settings. The results for 0-shot, 1-shot with long annotations, 3-shot (only for mixed annotations), and 10-shot are presented separately.

lang_pair	#shots	Unconstrained Decoding			Constrained Decoding		
		UPOS	UAS	LAS	UPOS	UAS	LAS
gl	0	82.44	33.15	23.92	91.22	44.79	38.55
gl_gl	1 (long)	73.38	39.84	30.61	92.07	46.31	**41.31**
	10	82.16	37.93	31.80	**93.70**	44.46	40.29
gl_es	1 (long)	83.74	**46.71**	34.16	91.28	45.69	40.12
	10	66.35	33.76	22.28	90.26	43.84	39.00
gl_pt	1 (long)	73.89	40.63	29.09	92.01	45.86	39.90
	10	66.80	36.69	27.29	92.40	46.15	40.97
gl_mixed	3	61.45	34.27	23.24	91.90	45.81	40.29
	10	75.46	34.27	25.55	92.12	45.86	41.02

Table 7: performance for Galician target language in 1-shot (long) and 10-shot settings; 3-shot results shown only for mixed annotations; scores in %

For the target language Catalan, we can also observe that the results in the unconstrained setting fluctuate greatly, as is the case with Galician. The lowest UPOS value for the language pair gl_es in the 5-shot scenario is only 55.71%, which is the lowest UPOS score overall, while the highest UPOS for this target language for the same language pair was achieved in the 1-shot (long) scenario. The UPOS values in constrained decoding only fluctuated between 90.26% (gl_es, 10-shot) and 93.70% (gl_gl, 10-shot). For the lowest UPOS score in the

unconstrained setting, this represents a performance increase of over 33%, from 55.71% to 90.83% (gl_es, 5-shot). The table also shows that for the target language Galician, all performance values could be increased by constrained decoding compared to unconstrained decoding, with one exception (gl_es, 1-shot (long), UAS). The LAS scores were greatly improved, as can be observed in gl_mixed in the 3-shot scenario (unconstrained decoding: 23.24%; constrained decoding: 40.29%). This is followed by consideration of the results for the target language Occitan. Table 8 shows the model performance for Occitan as the target language under different few-shot settings and language pairs (oc_oc, oc_fr, oc_ca, oc_mixed). The UPOS, UAS and LAS values are shown as percentages for unconstrained and constrained decoding. The results for 0-shot, 1-shot (long), 3-shot (mixed), and 10-shot are displayed separately.

lang_pair	#shots	Unconstrained Decoding			Constrained Decoding		
		UPOS	UAS	LAS	UPOS	UAS	LAS
oc	0	74.64	35.32	24.88	85.47	44.48	33.67
oc_oc	1 (long)	85.92	**49.27**	39.81	**89.64**	46.40	37.05
	10	83.98	45.39	**41.26**	89.08	47.07	38.18
oc_fr	1 (long)	77.43	43.81	33.13	**89.64**	46.40	37.05
	10	75.49	37.14	27.55	88.74	46.96	36.82
oc_ca	1 (long)	75.49	38.96	28.40	88.18	46.17	36.15
	10	75.61	40.41	30.95	87.50	44.26	34.12
oc_mixed	3	75.36	34.83	26.94	86.26	45.72	35.47
	10	77.55	37.86	28.76	88.18	47.75	37.50

Table 8: performance for Occitan target language in 1-shot (long) and 10-shot settings; 3-shot results shown only for mixed annotations; scores in %

The results for the target language Occitan once again show an improvement in performance using constrained decoding compared to the unconstrained setting, albeit with less pronounced differences than in Galician. Nevertheless, there are clear exceptions concerning the UAS. In the unconstrained setting, the UAS for the language pair oc_oc is higher than in the constrained setting for 1-shot (short), 1-shot (long), and 5-shot. In the unconstrained setting, performance clearly benefits from showing examples in the Occitan language, as high values were achieved in all metrics compared to the other unconstrained decoding results, and the UAS is also partially higher than with constrained decoding. The lowest UPOS value in the unconstrained decoding experiments is 56.31% (oc_ca, 3-shot), and the highest is 85.92% (oc_oc, 1-

shot (long)). Another interesting observation is that for the oc_mixed language pair in the constrained setting, the result values in all metrics increase continuously with the number of shots.

5.3 Results for high-resource Languages and Basque

The few-shot learning experiments for high-resource languages and Basque were conducted exclusively in the constrained setting. The following shows the results of the 1-shot (long) and 10-shot scenarios, and for mixed annotations, the results for 3-shot and 10-shot scenarios. The first table, table 9, shows the results for the target languages French and Spanish. Occitan was not included as a source language for the Spanish target language, which is why the corresponding field is left blank.

Target language: French					Target language: Spanish				
lang_pair	#shots	UPOS	UAS	LAS	lang_pair	#shots	UPOS	UAS	LAS
fr	0	95.73	48.91	43.13	es	0	92.56	43.73	36.29
fr_fr	1(long)	**95.90**	48.32	43.80	es_fr	1(long)	93.19	44.09	39.25
	10	95.81	49.92	44.05		10	93.19	44.89	38.71
fr_es	1(long)	94.22	48.24	43.30	es_es	1(long)	93.46	45.79	39.70
	10	94.39	47.82	42.80		10	93.55	45.25	39.25
fr_ca	1(long)	95.39	49.58	44.81	es_ca	1(long)	94.27	45.70	**40.50**
	10	95.56	48.74	43.38		10	93.55	45.61	39.16
fr_oc	1(long)	93.63	47.40	42.71					
	10	94.97	**50.25**	**45.14**					
fr_mixed	3	95.39	47.82	43.05	es_mixed	3	**94.35**	**46.15**	39.78
	10	95.23	50.17	44.81		10	93.55	46.06	39.43

Table 9: performance for French and Spanish target languages in 1-shot (long) and 10-shot settings; 3-shot results shown only for mixed annotations; scores in %

Very high UPOS values were achieved for the target language French, ranging from 93.47% (fr_es, 1-shot) to 95.90% (fr_fr, 1-shot (long)). The highest values for UAS and LAS were achieved with the source language Occitan in the 10-shot scenario: here, the UAS is 50.25% and the LAS is 45.14%. The fact that the highest result was achieved here is surprising for an under-resourced language such as Occitan. Again, higher shot numbers do not automatically lead to a higher performance score. Even though the highest UAS and LAS values were

achieved in the 10-shot scenarios for the language pairs fr_fr, fr_oc, and fr_mixed, no continuous improvement per higher shot number can be observed.

The results for the target language Spanish are similar. It can be observed that the result values for each metric generally fluctuate slightly more than for French. In addition, the LAS values achieved, which range between 36.29% (0-shot) and 40.50% (es_ca, 1-shot (long)), are considerably lower than for French. The UPOS values range from 91.58% (es_es, 1-shot (short)) to 94.35% (es_mixed, 3-shot).

In general, one would expect that the mixed annotations could lead to confusion during parsing, since they involve sentences in different languages that are subject to different grammatical rules. Nevertheless, this does not seem to be the case. Overall, the results for the target language French are highest, which is due to the fact that a French development script was used to develop the constraint decoding script, and the constraints were set for the French language. The results for Portuguese and the low-resource non-Romance language Basque are shown in table 10.

Target language: Portuguese					Target language: Basque				
lang_pair	#shots	UPOS	UAS	LAS	lang_pair	#shots	UPOS	UAS	LAS
pt	0	91.52	49.03	41.44	eu	0	76.29	35.05	15.46
pt_fr	1 (long)	92.08	47.58	42.25	eu_fr	1 (long)	76.63	38.83	17.70
	10	91.60	46.20	40.95		10	77.49	39.00	18.21
pt_es	1 (long)	92.08	46.93	42.65	eu_es	1 (long)	76.29	39.00	19.07
	10	91.28	46.28	41.36		10	80.07	37.80	17.70
pt_pt	1 (long)	93.13	47.98	42.57	eu_eu	1 (long)	84.36	**43.99**	21.13
	10	**93.78**	49.11	**44.02**		10	**85.57**	41.75	**21.31**
pt_gl	1 (long)	92.00	**49.27**	43.54	eu_oc	1 (long)	75.95	35.91	16.49
	10	92.41	46.45	41.68		10	73.37	35.74	17.53
pt_mixed	3	92.00	48.71	43.70	eu_mixed	3	76.63	37.46	18.38
	10	92.89	46.61	42.25		10	77.84	38.14	18.73

Table 10: performance for Portuguese and Basque target languages in 1-shot (long) and 10-shot settings; 3-shot results shown only for mixed annotations; scores in %

For the high-resource target language Portuguese, performance values similar to those for French or Spanish can be recorded. The UPOS values here range between 91.11% (pt_es, 5-shot) and 93.78% (pt_pt, 10-shot). Relatively high LAS and UAS values were also achieved, with the highest UAS at 49.27% (pt_gl, 1-shot (long)) and the highest LAS at 44.02% (pt_pt,

10-shot). It is also interesting to note that for the source languages Spanish, French, Galician, and mixed annotations, performance flattened out again in the 10-shot scenario compared to the previous scenarios, especially for UAS and LAS. While the results for the Romance languages are similar, the results for the non-Romance target language Basque are distinctly different. The UPOS values still provide satisfactory results, ranging from 72.16% (eu_es, 1-shot (short)) to 85.57% (eu_eu, 10-shot). With values of 34.36% (eu_es, 1-shot (short); eu_fr, 1-shot) and 43.99% (eu_eu, 1-shot (long)), the UAS is on average below the values for Romance target languages. That said, the LAS values, which range from 15.46% (0-shot) to 21.31% (eu_eu, 5-shot, and 10-shot), are considerably lower than the LAS values for the other target languages. It is also worth noting that Basque, as the source language, delivers the best results here. The reason for the poorer performance values for Basque could be that the constraints are adapted to Romance languages, and therefore, the grammatical rules may not be appropriate for Basque. The LAS for Basque in the unconstrained 0-shot setting was also very low at 17.01%, which shows that language models in general may have difficulty tagging and parsing the Basque language. Basque is a language isolate considered morphologically and syntactically complex, mainly due to its poly-personal verb inflection and ergative-absolutive sentence structure (Hualde & Ortiz De Urbina, 2003).

After presenting the results of the experiments, they are interpreted in the following section.

5.4 Interpretation and Implications of the Results

Analyzing the results of the baseline experiments revealed that performance often fluctuates greatly between 0-shot, 1-shot, and few-shot settings. In comparison, constrained decoding yielded more stable performance values. Contrary to the expectation that more example annotations would improve performance, this pattern could not be confirmed, meaning that H1 could not be supported.

The stability in constrained decoding suggests that structural and linguistic constraints already capture the essential linguistic patterns. Additional examples or annotations therefore only lead to minimal performance improvements. This might suggest that prompt engineering plays a less important role here. Another reason for the stable performance could be that Romance languages share a multitude of structural features, so the constraints designed for one language, in this case French, can be efficiently transferred to related languages. Constraints developed for high-resource languages like French could greatly benefit low-resource languages like Occitan.

It is particularly noteworthy that the 0-shot performance was only slightly below the few-shot performance values. This shows that minimal or no annotations are often sufficient to achieve promising results when appropriate constraints are in place. Large amounts of training data are therefore not necessary, which is particularly important for low-resource languages. Less data saves time and resources in collection, cleaning, and annotation. In addition, since few-shot experiments are more computationally intensive (Brown et al., 2020), obtaining comparable performance in 0-shot or 1-shot settings offers a considerable advantage.

The effects of constraints are also evident at the task level: particularly high scores were achieved for POS tagging (often above 90%), while more complex tasks such as syntactic parsing showed smaller improvements. Parsing is more demanding (De Marneffe et al., 2021), and constraints enable more targeted use of the linguistic knowledge available in the model. Recent studies (Raspanti et al., 2025) confirm that grammatical constraints improve syntactic correctness and parsing accuracy, where 0-shot experiments with constrained decoding deliver results that are close to those of a 5-shot scenario without constraints.

In summary, the results indicate that structural and linguistic constraints stabilize performance, that additional examples provide only limited benefit, and that low-resource languages can profit from constraints based on related high-resource languages. Even with minimal annotations, substantial time and effort can be saved without sacrificing performance compared to few-shot experiments. After presenting and interpreting the results, we will move on to a comprehensive analysis in the following chapter.

6. Analysis

This chapter comprises a series of quantitative and qualitative analysis steps that help to better classify and interpret the available results of the experiments. The quantitative analysis refers to the scores that were calculated when evaluating the output of the experiments, i.e. UPOS, LAS and UAS. The qualitative analysis, on the other hand, focused on the actual output, i.e. the annotated sentences, generated in the experiments. This includes the assignment of labels, errors in syntactic attachments and the correlation between similarity and model performance.

6.1 Quantitative Analysis

The quantitative analysis can be divided into two parts. First, a significance test was performed to determine whether the observed difference between two values is in fact statistically significant. This is important to be able to make statements about the improvement or deterioration

of model performance. A language similarity matrix was then created to test hypothesis H3 and a correlation test was then used to check whether there is a correlation between language similarity and model performance.

6.1.2 Significance Test

For the significance test, the statistical calculations were performed with `scipy.stats.norm`[34]. A Z-test for two proportions was used to test whether the difference of two parsing scores is statistically significant (Bortz & Schuster, 2011). The parsing results, i.e. UPOS, UAS and LAS, of the constrained decoding were to be systematically compared with each other. The aim was to determine whether the differences between the respective results are statistically significant.

Two basic types of comparison were carried out:

1. Comparisons across different shot scenarios within the same language pair (e.g. ca_ca: 0-shot vs. 1-shot).
2. Comparisons of identical shot scenarios across different language pairs, keeping the target language constant while varying the source language (e.g. 0-shot: ca_ca vs. ca_oc)

To determine significance, the absolute difference between the results was first calculated and then the p-value was determined. According to the null hypothesis, a difference is considered significant if $p \leq 0.05$ (Bortz & Schuster, 2011; Fahrmeir et al., 2023).

Cohen's h, which is a distance measure between two proportions, was also calculated in order to not only measure whether there is statistical significance, but also to be able to classify the significance of the difference. The effect sizes "negligible", "small", "medium" and "large" were used here, whereby only the effect sizes "small" and especially "negligible" were retained in the calculations. This means that even if there is a statistically significant difference, the different shot settings and different source languages for the same target language have little effect on parsing performance in practice, at least if the languages come from the same language family (Cohen, 2013). Even though these are not highly significant differences, some interesting patterns were discovered using this significance test. The results were analyzed in relation to the different metrics, UPOS, UAS and LAS. The exact results and the p-values can be viewed in the appendix.

[34] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.norm.html

The UPOS score was analyzed first. The language pair eu_eu showed the greatest significant improvement in the UPOS score, because here the performance could be enhanced from UPOS_0-shot = 76.29% to UPOS_10-shot = 85.57% (p-value ≈ 0.000056). In the Romance languages, the greatest significant difference is found in the language pairs oc_oc and oc_fr, where UPOS_0-shot = 85.47% and UPOS_1-shot (long) = 89.64%. Of the significant differences, 38.1% showed a significant improvement from 0-shot or 1-shot(short) settings to 10-shot settings. The highest UPOS scores were achieved by the minority language pair ca_ca and the high-resource language pair fr_fr, the former with UPOS = 95.59% and the latter with 95.81%. If we compare the UPOS scores at the level of language pairs, the results reveal that Basque as the source language for Basque as the target language delivers significantly better results than French, Spanish, Occitan or mixed annotations in the 3-, 5- and 10-shot scenarios. For the target language Catalan, example annotations in Catalan also delivered significantly better results than Spanish, Occitan or mixed example annotations in the 3-, 5- and 10-shot scenarios. For the target language Portuguese, the source language Portuguese also produced significantly higher UPOS scores than the source languages French and Spanish in the 10-shot scenario. For the target language Galician, the source languages Portuguese and Galician delivered significantly higher UPOS scores than the source language Spanish in the 10-shot scenario. With one exception, the UPOS value was the only metric that showed significant results when comparing language pairs under the same shot count scenario. The exception was the UAS in the 10-shot scenario, which was significantly higher for the language pair eu_eu than for the language pair eu_oc ($p \approx 0.04$).

If we look at statistically significant differences based on the shot counts, we observe less consistent patterns in the UAS score. The language pair eu_eu shows the greatest significant differences. Here, significant improvements in the UAS from 0-shot and 1-shot to 1-shot (long) and 0-shot to 3-, 5- and 10-shot could be recognized. For the language pair fr_oc there is a significant improvement from 1-shot (short) to 10-shot; for the language pair pt_es from 0-shot to 1-shot (short). Now we come to the LAS. The largest significant improvements in LAS can be observed for the language pair eu_eu, from 0-shot to 1-shot (long), 5-shot and 10-shot. The language pair gl_gl also shows significant improvements in LAS from 1-shot (short) to 1-shot (long), 3-shot and 5-shot. There is also a significant improvement in LAS for the language pair es_ca from 0-shot to 1-shot (long), for ca_ca from 1-shot (short) to 1-shot (long) and for ca_mixed from 0-shot to 10-shot. The significance test shows that the values for Basque are generally lower, but that both increasing shot counts and the use of Basque as a source language

in contrast to Romance languages lead to the greatest absolute improvements being achieved in the few-shot experiments.

6.1.3 Language Similarity and Correlation

To test hypothesis H3, first the similarity between languages was calculated, and afterwards the correlation between this similarity and the annotation performance of the model was computed. To calculate the similarity between the different languages, lang2vec[35] was used. Lang2vec is a collection of vector representations designed to make similarities and differences between languages measurable. Entire languages are to be represented as vectors. Lang2vec is based on the language database URIEL (Universal Research Index of Empirical Linguistic), which makes structural, genealogical and geographical information about languages available in vector form (Littell et al., 2017). Uriel collects data from different databases such as WALS (Dryer et al., 2024), Ethnologue (Eberhard et al., 2023) or Glottolog (Hammarström et al., 2025). Different feature sets can be loaded and used for calculation, such as syntax, phonology, inventory, fam and geo (Littell et al., 2017). The syntax, geo and fam features were used for this research. These features were chosen because there was sufficient language data available. For phonology and inventory, there was no data for Occitan and Galician (these vectors were empty). The nine languages to be analyzed were French, Spanish, Portuguese, Catalan, Occitan, Galician, Basque, Italian and Romanian. The linguistic features were then extracted. *Syntax* stands for the syntactic features, *fam* for language family relationships and *geo* for geographic proximity (Littell et al., 2017). The similarity matrix was then calculated. For this purpose, the feature vectors were converted into `numpy`[36] arrays and then the cosine similarity was calculated. The results for each individual feature were output as a table showing the similarity values (s) between the language pairs. Finally, the average matrix of all features was calculated and an average score for all three features was calculated for each language pair. The result is shown as follows:

[35] https://github.com/antonisa/lang2vec

[36] https://numpy.org/

lang	fr	es	pt	ca	oc	gl	eu	it	ro
fr	**1.000**	0.856	0.873	0.868	0.901	0.852	0.525	0.825	0.796
es	0.856	**1.000**	0.969	0.948	0.933	0.948	0.550	0.883	0.827
pt	0.873	0.969	**1.000**	0.954	0.936	0.950	0.542	0.889	0.833
ca	0.868	0.948	0.954	**1.000**	0.936	0.929	0.523	0.893	0.853
oc	0.901	0.933	0.936	0.936	**1.000**	0.927	0.582	0.866	0.871
gl	0.852	0.948	0.950	0.929	0.927	**1.000**	0.454	0.856	0.833
eu	0.525	0.550	0.542	0.523	0.582	0.454	**1.000**	0.530	0.527
it	0.825	0.883	0.889	0.893	0.866	0.856	0.530	**1.000**	0.859
ro	0.796	0.827	0.833	0.853	0.871	0.833	0.527	0.859	**1.000**

Table 11: similarity scores for each language pair; scores in %

These average values were then also used for the correlation analysis. To calculate the correlation between language similarity and model performance, the `scipy.stats`[37] Python module was used to compute the Pearson and Spearman correlation. The Pearson correlation measures linear relationships. It is sensitive to the actual numerical values and is therefore ideal for detecting proportional changes, in the case of whether similarity values lead directly to an improvement in performance (Pearson, 1896). In contrast, the Spearman correlation measures monotonic relationships based on data ranks and not based on raw values, which makes it robust against outliers. This allows constant patterns to be recognized, even if the relationship is not entirely linear (Spearman, 1904). If both correlation values are similar, there is a linear relationship, if the Spearman coefficient is higher, this may indicate non-linear but consistent patterns. It is therefore important to consider both types of correlation in order to be able to make a differentiated statement here. All language pairs were considered individually and the correlations between each metric (UPOS, LAS and UAS) and language similarity score were compared. Since two correlation methods were calculated for each of three metrics (UPOS, LAS and UAS) for each of seven target languages, this yielded a total of 42 correlation values. The `scipy.stats`[38] module was then used to automatically test whether the relationship was statistically significant.

[37] https://docs.scipy.org/doc/scipy/reference/stats.html

[38] https://docs.scipy.org/doc/scipy/reference/stats.html

The correlation test showed that there was a strong correlation between the similarity value (s) and higher UPOS for the target language Catalan. Nevertheless, no correlation was found between UAS and LAS and language similarity. Specifically, this means that for the target language Catalan, the source languages Catalan (s = 1), Occitan (s = 0.936) and Spanish (s = 0.948) led to the highest UPOS values. For the target language Galician, there was a moderate correlation between the UPOS and the language similarity. The highest UPOS can be achieved with source languages Galician (s = 1), Portuguese (s = 0.950) and Spanish (s = 0.948). It is therefore advisable to use the same language as target and source language for Galician and Catalan. For Portuguese, the correlation between language similarity and model performance was weak and not statistically significant, while no significant correlations were observed for the other Romance languages (French, Spanish, and Occitan). For Basque, there was a strong correlation between UPOS and language similarity, and a moderate correlation between language similarity and LAS and UAS. Since only five significant correlations were identified in the 24 tests, amounting to 20.8%, hypothesis H3 can generally be rejected, except for individual languages and settings. In this context, it is worth noting that the correlation between performance and linguistic similarity was stronger in Basque, suggesting that this hypothesis might apply more to languages from different language families or language isolates, and that correlations are more difficult to detect with high similarity scores such as those found for Romance languages.

6.2 Qualitative Analysis

In the first step of the qualitative analysis, the most frequent errors in the assignment of labels were investigated, also providing information about the errors in the assignment of syntactic attachments. In the second step, hypothesis H2 was tested by computing the syntactic similarity between example and target sentences and examining the relationship between sentence similarity and model performance.

6.2.1 Most Common Errors

In order to analyze the output on a qualitative level, a Python script was developed using the `conllu`[39] library for parsing CoNLL-U formatted files and `collections`[40] for generating error statistics. Based on the linguistic annotations, it compares the system file (i.e. the output

[39] https://pypi.org/project/conllu/

[40] https://docs.python.org/3/library/collections.html

files of the in-context learning experiments) with the gold standard and identifies three error types, UPOS, HEAD and DEPREL. The script identifies shot scenarios and language pairs, analyzes average error rates per language pair and identifies the most common error-causing words for each pairing. The results are grouped by scenario so that differences between different language pairs can be analyzed. The results of this error analysis are presented below, showing the average errors in all shot scenarios for each language pair, in the order `correct label → wrong label`.

Target lang	source lang	Error types
fr	non	UPOS: `ADV → PART; X → PROPN` DEPREL: `case → mark; punct → dep`
	fr, es, ca, mixed	UPOS: `AUX → VERB; ADV → PART` DEPREL: `case → mark; nmod → obl`
	oc	UPOS: `AUX → VERB; ADV → PART` DEPREL: `nmod → obl; obl:mod → obl`
es	non	UPOS: `PROPN → NOUN; SCONJ → PRON` DEPREL: `nmod → obl; case → mark`
	ca	UPOS: `PROPN → NOUN; SCONJ → PRON` DEPREL: `case → mark; nmod → obl`
	es, fr	UPOS: `PROPN → NOUN; AUX → VERB` DEPREL: `case → mark; nmod → obl`
	mixed	UPOS: `PROPN → NOUN; AUX → VERB` DEPREL: `nmod → obl; case → mark`
pt	non	UPOS: `CCONJ → SCONJ; VERB → AUX` DEPREL: `nmod → obl; punct → dep`
	es	UPOS: `AUX → VERB; CCONJ → SCONJ` DEPREL: `nmod → obl; case → mark`
	pt, fr, mixed	UPOS: `CCONJ → SCONJ; PROPN → NOUN` DEPREL: `nmod → obl; case → mark`
	gl	UPOS: `CCONJ → SCONJ; AUX → VERB` DEPREL: `nmod → obl; case → mark`

Table 12: most common errors for high-resource target languages

In this and the following tables, the two most frequent UPOS and DEPREL errors are presented, depending on the target and source language, ordered from left to right, from most frequent error to second most frequent error. It is interesting to note that the source language has an impact on the type of errors in the annotations. Even if the types of errors are partly the same, they differ in order, i.e. in frequency, but sometimes the same UPOS and DEPREL errors occur for different source languages: For example, for the target language French, the source languages French, Spanish, Catalan and mixed annotations lead to the same types of errors. This also applies to the source languages Spanish and French for the target language Spanish and the source languages Portuguese and French, as well as mixed annotations, for the target language Portuguese.

One of the most common errors across all target languages in UPOS is the incorrect assignment of `AUX → VERB`. This means that auxiliary verbs are not reliably recognized and are often classified as `VERB`s. For the target language Spanish, `PROPN → NOUN` and `SCONJ → PRON` errors also occur frequently. The former means that proper nouns are often incorrectly classified as nouns, while the latter means that superordinate conjunctions are incorrectly classified as pronouns. In Spanish, the word *que* most frequently leads to `SCONJ → PRON` errors, as it means *that* and is a subordinating conjunction, but can also be used as a relative pronoun (`PRON`). Correct classification can be difficult, especially in syntactically complex sentences. For the target language Portuguese, `CCONJ → SCONJ` is one of the most common errors, i.e. when a coordinating conjunction is incorrectly classified as a subordinating conjunction. Again, the word *que* often leads to misclassifications. Considering the most frequent errors in the dependency relations (DEPREL), it can be observed that the error types `nmod → obl` and `case → mark` occur most frequently across all language pairs. An example of a `nmod → obl` error shows in es_es_10 in sentence 3: “El Acta de la Independencia de Guayaquil, está escrita en el fuste de la columna.“ (English: “The Act of Independence of Guayaquil is written on the shaft of the column. “) In this case, *fuste* is annotated as `nmod` of *acta*. In fact, *fuste* should be `obl` of *escrita*. *Fuste* is not a nominal modifier here, but an adverbial determiner of the predicate; the superordinate verbal reference is disregarded here. Syntactically, the sentence is structured like a nominal phrase, but the semantic context shows that *fuste* is the place of the action, i.e. the local complement. When considering the `case → mark` error types, it is noticeable that words such as *de*, *en* and *à* often lead to problems. This is shown in sentence 13 in the output file of fr_fr in the 5-shot scenario: “On retrouve ici une touche très melvillienne et qui à le lieu de souligner la thématique de le livre y ajoute du sens.“ (English: “Here we find a very Melvillian touch, which, instead of emphasizing the theme of the book, adds meaning to it.“)

De was annotated here as a `mark` of *souligner*, but is actually the `case` of *livre*. This could indicate that the model, despite detailed treatment in the constrained decoding, has difficulties with the annotation and correct assignment of contractions (multiword tokens). It is also interesting to note that the model does not recognize that it has already assigned the *de* that precedes *souligner* as a `mark` of *souligner*. For the language pair fr_oc, the `obl:mod → obl` error type is the second most frequent and for the target languages French and Portuguese, the `punct → dep` error type is the most frequent in the 0-shot scenarios.

Table 13 shows the most common error types for the Romance minority languages.

target lang	source lang	error types
ca	non	UPOS: `PROPN → NOUN; PROPN → ADJ` DEPREL: `flat → nmod; nmod → obl`
	es	UPOS: `PROPN → NOUN; AUX → VERB` DEPREL: `flat → nmod; case → mark`
	ca, oc, mixed	UPOS: `PROPN → NOUN; AUX → VERB` DEPREL: `flat → nmod; nmod → obl`
gl	non	UPOS: `PRON → SCONJ; AUX → VERB` DEPREL: `case → mark; nmod → obl`
	gl	UPOS: `AUX → VERB; ADJ → NOUN` DEPREL: `nmod → obl; ccomp → acl`
	pt	UPOS: `AUX → VERB; PRON → SCONJ` DEPREL: `nmod → obl; case → mark`
	es, mixed	UPOS: `PRON → SCONJ; AUX → VERB` DEPREL: `nmod → obl; case → mark`

Table 13: most common errors for minority target

target lang	source lang	error types
oc	non	UPOS: VERB → AUX; ADV → PRON DEPREL: nmod → obl; obj → nsubj
	oc	UPOS: VERB → AUX; PRON → DET DEPREL: nmod → obl; obj → nsubj
	fr	UPOS: VERB → AUX; PRON → SCONJ DEPREL: nmod → obl; case → mark
	ca	UPOS: VERB → AUX; PRON → DET DEPREL: case → mark; nmod → obl
	mixed	UPOS: VERB → AUX; PRON → DET DEPREL: nmod → obl; case → mark

Table 13 (continued): most common errors for minority target

Looking at the most frequent UPOS errors for the Romance minority languages, it can be observed that for the target language Catalan, as for Spanish, the error type PROPN → NOUN occurs most frequently across all source languages. For the target language Catalan, proper nouns are not only incorrectly classified as nouns, but also as adjectives, which is the most frequent UPOS error here. As with the higher resource languages, the assignment of auxiliary verb and verb also leads to errors here, as can be seen from the frequency of the error types VERB → AUX and AUX → VERB in the minority languages. Other error types that occur frequently concern the assignment of PRON, such as PRON → SCONJ (gl_0, gl_pt, gl_es, gl_mixed, oc_fr), PRON → DET (oc_oc, oc_ca and oc_mixed) and ADV → PRON (oc_0). Among the DEPREL errors, the nmod → obl and case → mark errors are also among the most common errors, as is the case for high-resource languages. In addition, the most common error type for the target language Catalan is flat → nmod. For oc_0 and oc_oc, the error type obj → nsubj is the second most common and for gl_gl ccomp → acl. UPOS errors and DEPREL errors can have a direct correlation, for example with the error types PROPN → NOUN and flat → nmod, as shown in sentence 46 in the Catalan dataset: “Antoni Castells, Doctor en Medicina i Cirurgia per la Universitat de Barcelona, és Catedràtic de Dermatologia de la Facultat de Medicina de la Universitat Autònoma de Barcelona.” (English: “Antoni Castells, Doctor in Medicine and Surgery from the University of Barcelona, is Professor of Dermatology at the Faculty of Medicine of the Autonomous University of Barcelona.“) In this instance, *Doctor* was incorrectly identified as NOUN instead of PROPN, which meant that

Doctor could not be recognized as a `flat` of *Antoni*, but was classified as an apposition to *Universitat*. Without a correct POS classification as PROPN, the model has greater difficulty recognizing the stringing together of words as multiword names (such as institutional names like *Universitat Autònoma de Barcelona*).

Table 14 presents the most common error types observed for Basque.

target lang	source lang	error types
eu	non	UPOS: `AUX → VERB; ADJ → NOUN` DEPREL: `aux → conj; obl → nsubj`
	es, fr, mixed	UPOS: `AUX → VERB; ADJ → NOUN` DEPREL: `aux → conj; aux → xcomp`
	eu	UPOS: `AUX → VERB; VERB → AUX` DEPREL: `aux → conj; obj → nsubj`
	oc	UPOS: `AUX → VERB; ADJ → NOUN` DEPREL: `aux → conj; obj → nsubj`

Table 14: most common errors for Basque as target

The most frequent errors for the non-Romance target language Basque are examined individually here. For all source languages, the misannotation of auxiliary verbs as full verbs is the most frequent UPOS error type, and the misannotation of auxiliaries as conjuncts is the most frequent DEPREL error type. Sentence 1 in the Basque dataset can be used to illustrate these errors: “Atenasen ordea, beste bost jarduera gehiago izan daitezke.” (English: “In Athens, however, there may be five more activities.”) In this case, *daitezke* was incorrectly annotated as a `VERB` and conjunct of *izan* in the 0-shot scenario, although it is an auxiliary of *izan*. *Izan daitezke* can be translated as *may be* or *could be*. The second most common UPOS error type for the source languages Spanish, French, Occitan, as well as mixed annotations and 0-shot, is the misclassification of adjectives as nouns. Basque is an agglutinative language (Hualde & Ortiz De Urbina, 2003), unlike the Romance languages, which are inflectional (Pöckl et al., 2013), which means that each morphological unit necessary for the different functions, including case, is added independently to the stem of the word. Affixes for article, number and case are attached to the word in this order. These markers appear at the last element of the noun phrase. This can be a noun, but also an adjective or determiner. Because adjectives and nouns often have the same suffixes and word order is relatively free in Basque, it is easy to misclassify an adjective as a noun (Aduriz et al., 1995). This type of error can be observed in sentence 5 (eu_0): “Aurten izan ditudan aukera apurrak ondo baliatu ditudala uste dut.” (English: “I believe

that I have made good use of the few opportunities I had this year.") In this case, *apurrak* (*a few*) was wrongly classified as a noun, although it is an adjective in the context of *aukera apurrak* (*the few opportunities*). With Basque as the source language, the misclassification of full verbs as auxiliary verbs is the second most common type of error. In the output of eu_eu in the 1-shot scenario, we can find this type of error in sentence 17: "Aurrelari langilea, azkarra eta indartsua dela erakutsi du denboraldi-aurrean." (English: "He has shown during the pre-season that he is a hardworking, fast, and strong striker.") The word *dela* was annotated here as `AUX` but remains the only verb with a predicative function in the subordinate clause. The second most frequent DEPREL errors for the language pairs eu_oc and eu_eu are the misclassifications of `obj` as `nsubj`. The second most frequent DEPREL error in the 0-shot scenario is the `obl` → `nsubj` error type. The language pairs eu_es, eu_fr and eu_mixed show the misclassification of `aux` as `xcomp` (open clausal complement) as the second most frequent DEPREL error.

To summarize, it can be said that different source languages cause different types of errors for the same target language. The misclassification of auxiliaries as full verbs and vice versa is a common error for high-resource Romance languages, Romance minority languages and also Basque. For both high-resource Romance languages and minority languages, `case` → `mark` errors are very common, as are `nmod` → `obl` errors. Basque is the only target language for which the `ADJ` → `NOUN` error is one of the most frequent error types. Overall, the results show that error patterns in in-context learning differ systematically depending on the target and source language. This is particularly true for structurally unrelated target languages such as Basque, where different syntactic structures lead to different error patterns.

6.2.2 Syntactic Similarity and Model Performance

The last step of the qualitative analysis was to examine whether the syntactic similarity of an example sentence to a target sentence leads to a successful annotation by the model. For this purpose, we investigated whether syntactic similarity between example annotation sentences and target sentences influences parsing performance in 1-shot scenarios (standard, short, and long). To this end, a direct comparison approach was developed that evaluates the differences in parsing accuracy between sentences with high and low syntactic similarity to available examples and shows in which languages syntactic similarity offers meaningful advantages.

The analysis workflow processed files in CoNLL-U format: the gold standard annotations, example sentences, and system outputs in seven languages (Catalan, Spanish, Basque, French, Galician, Occitan, and Portuguese). The system automatically identified the three 1-shot parsing scenarios, namely 1-shot, 1-shot (short), and 1-shot (long).

The syntactic similarity calculation employed a hybrid approach combining three complementary measures: TF-IDF cosine similarity (40%), sequence matching (40%) and n-gram overlap (20%). Term Frequency-Inverse Document Frequency (TF-IDF) is mainly used to recognize the importance of account words in documents. Nevertheless, it can also be used to calculate textual similarity (Albitar et al., 2014). First, syntactic signatures were generated from each sentence in the example annotations file and the gold file, looking like this:

example_annotation_oc_1S_short.conllu
text = Es lo professional de lo tropelet.
VERB DET NOUN ADP DET NOUN PUNCT root det nsubj case det nmod punct root(ROOT→ VERB) **det(NOUN→ DET) nsubj(VERB→ NOUN) case(NOUN→ ADP) det(NOUN→ DET) nmod(NOUN→ NOUN)** punct(VERB→ PUNCT) LEN_5

gold_file_oc; sent_id = 32
text = Au bon moment, lo hilh deu rei se botèc la pluma dens la boca.
ADP DET ADJ NOUN PUNCT DET NOUN ADP DET NOUN PRON VERB DET NOUN ADP DET NOUN PUNCT case det amod obl punct det nsubj case det nmod expl root obj case det obl punct case(NOUN→ ADP) det(NOUN→ DET) amod(NOUN→ ADJ) obl(VERB→ NOUN) punct(VERB→ PUNCT) **det(NOUN→ DET) nsubj(VERB→ NOUN) case(NOUN→ ADP) det(NOUN→ DET) nmod(NOUN→ NOUN)** expl(VERB→ PRON) root(ROOT→ VERB) obj(VERB→ NOUN) case(NOUN→ ADP) det(NOUN→ DET) obl(VERB→ NOUN) punct(VERB→ PUNCT) LEN_15
Similarity score ≈ 0.50

Vectors were subsequently generated for these signatures. Since the search was not for similar words but for similar syntactic structures, it was not the words that were given positions but the labels. In this case, the TF-IDF matrix worked with syntactic labels as features instead of lexical words. The cosine similarity of these vectors was then calculated, and the maximum similar gold sentence was found for each example sentence (Jurafsky & Martin, 2025). Since TF-IDF solely collects the frequency of syntactic elements, this method was supplemented with sequence detection, i.e., the order in the sentence structure, as well as with an n-gram method that detects local syntactic patterns. The TF-IDF method (40%) captured what occurs in a sentence, while sequence matching (40%) and n-gram overlap (20%) captured the sentence

structure. The `SequenceMatcher` from the Python module `difflib`[41] was used for sequence matching, which is based on the Ratcliff-Obershelp algorithm (Ratcliff & Metzener, 1988). This served to find the longest common subsequences between the token lists. Bigrams were then created from both signatures, from which the Jaccard index was calculated (Jaccard, 1912). This captured local structural similarities (Jurafsky & Martin, 2025). The values of the different approaches were then combined into a hybrid similarity score.

Initially, the script calculated the similarity scores between all example sentences and each gold sentence using the hybrid similarity method. For each gold sentence, the maximum similarity of all available examples was determined, giving each gold sentence a single similarity score that indicated how well it matched the best available example. These similarity scores were then used to divide the gold sentences into "random" (≤ median similarity) and "similar" (> median similarity). Next, the corresponding annotated sentences were assigned from the model output, and the gold standard was compared with the system output token by token, allowing UPOS, UAS, and LAS to be calculated for each sentence. Consistently, 50 sentences per file were processed in the analysis, evenly divided into 25 "random" and 25 "similar" group members. This created separate performance arrays for each metric for both groups, "random" and "similar". The arrays contained all individual sentence scores for a group, and the per-file mean was then calculated for each metric: UPOS, UAS and LAS. Each file was analyzed independently, generating separate averages (`random_las`, `similar_las`, etc.). These values formed the basis for comparison and subsequent statistical tests. The analysis covered a total of 3,150 sentences across 63 files, spread across seven languages with varying data density: Catalan, Spanish, Galician and Occitan with 9 files each (450 sentences per language), Basque, French, and Portuguese with 12 files each (600 sentences per language). Each file consisted of 50 sentences, which were evenly distributed between random (25 sentences) and similar groups (25 sentences).

The mean performance scores for the similar group and the random group for UPOS, UAS, and LAS are listed in Appendix Table A.8 for all 1-shot scenarios for all language pairs, including their differences. For better comprehensibility and readability, average values were calculated for the target languages. These values are used in the following descriptive analysis. Table 15 shows the UPOS, LAS, and UAS average values for each target language, including the difference Δ between the similar and random scores in absolute numbers.

[41] https://docs.python.org/3/library/difflib.html

target_lang	UPOS			LAS			UAS		
	R	**S**	**Δ**	**R**	**S**	**Δ**	**R**	**S**	**Δ**
ca	0.931	0.926	-0.004	0.395	0.39	-0.005	0.477	0.468	-0.009
es	0.902	0.937	0.035	0.406	0.423	0.017	0.478	0.494	0.016
eu	0.757	0.756	0	0.184	0.212	0.028	0.401	0.433	0.031
fr	0.928	0.948	0.021	0.41	0.443	0.033	0.498	0.521	0.023
gl	0.912	0.916	0.004	0.398	0.409	0.011	0.454	0.471	0.017
oc	0.847	0.88	0.033	0.397	0.399	0.002	0.544	0.51	-0.034
pt	0.899	0.924	0.026	0.407	0.447	0.04	0.483	0.511	0.027

Table 15: mean performance scores across target languages (Δ = Similar – Random)

Looking at the average values for each target language, it can be seen that the influence of syntactic similarity varies depending on the target language. A positive difference indicates that the similar group achieves better results than the random group, which indicates a positive influence of syntactic similarity. A negative difference, on the other hand, means that random sentences yield better results and thus no clear advantage due to syntactic similarity can be identified. As can be seen, the target languages Portuguese and French show consistently positive differences, while Catalan shows slightly negative differences. Occitan shows a negative difference for UAS, but otherwise only positive differences. Basque also shows positive differences for LAS and UAS, but no difference for UPOS. Galician and Spanish show small but positive trends. The trends are clearly visible in Figure 5, which shows the relative differences (Δ) in percent.

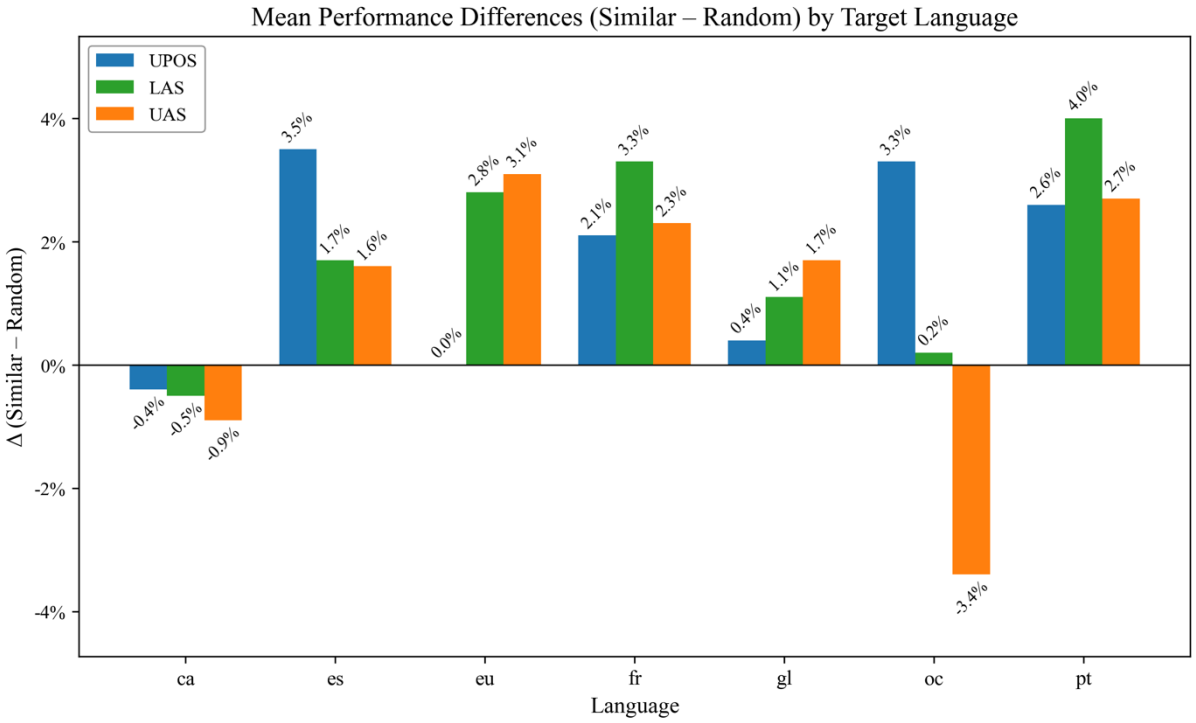


Figure 5: mean performance differences (Δ = Similar – Random, %) by target language and metric (UPOS, LAS, UAS)

As can be seen in Figure 5, on average, Portuguese shows the most consistent improvements across all three metrics, with positive differences between 2.6% and 4%. Occitan benefits from POS tagging (3.3%), but the positive difference is very small for LAS (0.2%) and negative for UAS (-3.4%). French shows moderate average improvements, with mean differences ranging from 2.1% to 3.3%. Catalan shows negative difference values, which suggests that syntactic similarity does not provide any advantage here. For Basque, syntactic similarity leads to better results in parsing, but not in tagging.

Building on the descriptive results, the next step was to examine whether the differences between the similar and random groups were statistically significant. Therefore, the "similar" group score was compared with the "random" group score per metric using an independent samples t-test. In addition, Cohen's d was used as an effect size to measure practical significance. To confirm the hypothesis, the similar group had to outperform the random group, i.e., the model performance had to be higher. Furthermore, the result had to be statistically significant, i.e., $p \leq 0.05$, with Cohen's $d \geq 0.2$ to represent at least a small effect (Cohen, 2013). The support rate was calculated accordingly and indicated the percentage of files in a target language that met all three criteria.

The results are presented below, organized by target language. The following graph (figure 6) shows the support rates, i.e., the proportion of cases in which the hypothesis is confirmed across the three metrics.

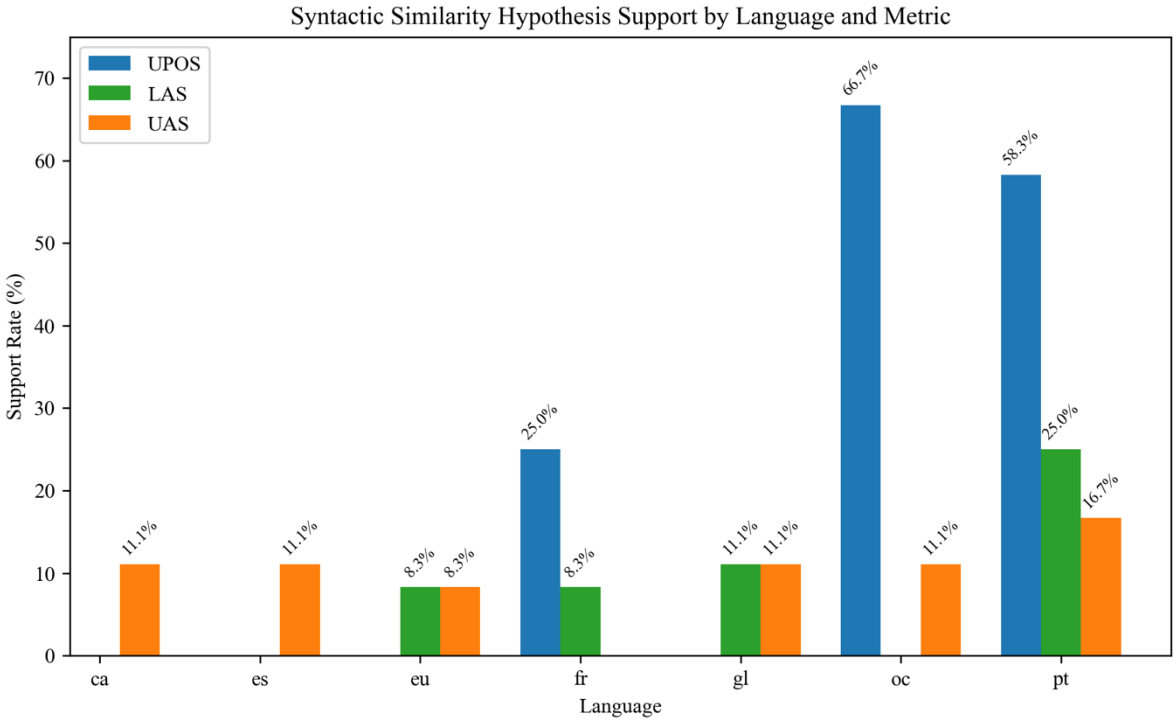


Figure 6: support rates (in %) for syntactic similarity hypothesis across seven target languages by three metrics (UPOS, LAS, UAS)

According to this analysis, Portuguese is the language in which syntactic similarity and model performance are most closely related, and it showed the most consistent results. For UPOS tagging, the support rate for the hypothesis was 58.3%, for LAS it was 25.0%, and for UAS it was 16.7%. Occitan also showed a very high support rate with UPOS tagging (66.7%), but with dependency parsing, only 11.1% support was achieved with UAS and no support at all with LAS. For French, the hypothesis was supported in 25% of cases with POS tagging, but only marginally with dependency parsing (LAS = 8.3%, UAS = 0%). For Galician, Catalan, Spanish, and Basque, no correlation was found between increasing UPOS and syntactic similarity. For Basque, the support rate for both UAS and LAS was 8.3%. For Galician, the hypothesis was confirmed for UAS and LAS in 11.1% of cases. For Catalan and Spanish, the rate for LAS was also 11.1%. The significance tests thus complement the descriptive analysis by confirming some of the previously identifiable trends (e.g., in Portuguese), while in other cases they show that the differences are not statistically reliable (e.g., UAS in French).

Even though the analyses reveal some positive correlation between syntactic similarity and model performance, hypothesis H2 cannot be clearly confirmed. Across all languages and metrics, the correlations are inconsistent, although Portuguese, as the most robust case, showed clear improvements at all levels. Occitan and Galician show selective effects, with support limited to certain metrics. French, Catalan, and Spanish showed only marginal gains. Finally, Basque, as a non-Romance language, yields very low support values, which may reflect the impact of typological distance. Overall, it can be concluded that syntactic similarity is a relevant but not a universal factor for in-context learning.

The investigation of the correlation between syntactic sentence similarity and model performance was the final step in this series of analyses. Having analyzed the data and discussed the key findings, the following section will summarize the results into a comprehensive conclusion.

7. Conclusion

In this thesis, the in-context learning capabilities of a generative language model, Aya Expanse 32B, were tested in syntactic parsing. Romance languages, especially Romance minority languages, as well as Basque were examined. For in-context learning, free generation as well as constrained decoding strategies were applied. The aim of this thesis was to find out how different characteristics such as quantity of annotations, linguistic diversity, language similarity, and syntactic similarity in the example annotations affect model performance in in-context learning.

It was shown that a higher quantity of examples does not necessarily lead to higher performance values. Nevertheless, results from the unconstrained setting often showed improvements in performance from 0-shot to 1-shot, whether long or short. POS tagging benefited from structural constraints, which was not the case for UAS and LAS. Constrained decoding with structural and linguistic constraints generally resulted in stable performance values, i.e., less variation in performance values between the different shot scenarios, especially in comparison to the unconstrained decoding. This leads to the conclusion that LLMs benefit from structural and linguistic constraints in syntactic parsing, where the additional display of annotation examples has only a minimal effect. Furthermore, it was found that constraints developed for a high-resource language, in this case French, also improve performance for minority languages of the same language family. Constrained decoding could therefore be an effective way to achieve cross-lingual transfer, and large amounts of data for model training, fine-tuning, or in-context learning examples may be less relevant for high model performance.

Looking at the results of in-context learning experiments with mixed annotations, these showed stability in performance with constrained decoding but never led to significantly better results than examples in one language. The hypothesis that language similarity between source and target languages and model performance are related could generally be refuted, except for individual languages and settings. For Catalan and Galician, a correlation was found between high UPOS and language similarity. For Basque, the correlation between model performance and language similarity was higher, showing that the hypothesis might be more applicable to languages from different language families and language isolates.

A more detailed qualitative analysis of the output revealed that different source languages often lead to similar types of errors in a target language, but sometimes also produce differing types of errors. This demonstrates that, despite similar performance values, the generated output can often vary widely.

When investigating whether syntactic similarity between example annotations and the sentences to be annotated is related to model performance, inconsistent correlations were generally found. Nevertheless, the results also showed that syntactic similarity can play a role for some languages and specific tasks. For Portuguese, the strongest correlation between syntactic similarity and model performance was observed. Occitan showed a strong correlation for UPOS tagging. For French, the hypothesis for UPOS was confirmed in 25% of cases. For the other languages, only minor correlations were found.

This thesis aimed to show the potential contribution of computational linguistics to traditional philological fields, in this case Romance studies. It also sought to present approaches

to dealing with minority languages that do not involve large amounts of data, lengthy annotation processes, or model training. A limitation of these experiments is the use of small datasets of 50 sentences per dataset. Further research opportunities would be to continue the experiments with larger datasets or other minority languages and low-resource languages, especially languages from other language families.

Bibliography

Aakanksha, Ahmadian, A., Ermis, B., Goldfarb-Tarrant, S., Kreutzer, J., Fadaee, M., & Hooker, S. (2024). *The Multilingual Alignment Prism: Aligning Global and Local Preferences to Reduce Harm* (No. arXiv:2406.18682). arXiv. https://doi.org/10.48550/arXiv.2406.18682

Aakanksha, Ahmadian, A., Goldfarb-Tarrant, S., Ermis, B., Fadaee, M., & Hooker, S. (2024). *Mix Data or Merge Models? Optimizing for Diverse Multi-Task Learning* (No. arXiv:2410.10801). arXiv. https://doi.org/10.48550/arXiv.2410.10801

Aduriz, I., Alegria, I., Arriola, J. M., Artola, X., A, D. de I., Ezeiza, N., Gojenola, K., & Maritxalar, M. (1995). *Different Issues in the Design of a Lemmatizer/Tagger for Basque* (No. arXiv:cmp-lg/9503020). arXiv. https://doi.org/10.48550/arXiv.cmp-lg/9503020

Agerri, R., Gómez Guinovart, X., Rigau, G., & Solla Portela, M. A. (2018). Developing New Linguistic Resources and Tools for the Galician Language. In N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, & T. Tokunaga (Eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA). https://aclanthology.org/L18-1367/

Ahia, O., Kumar, S., Gonen, H., Kasai, J., Mortensen, D. R., Smith, N. A., & Tsvetkov, Y. (2023). *Do All Languages Cost the Same? Tokenization in the Era of Commercial Language Models* (No. arXiv:2305.13707). arXiv. https://doi.org/10.48550/arXiv.2305.13707

Albitar, S., Fournier, S., & Espinasse, B. (2014). An Effective TF/IDF-Based Text-to-Text Semantic Similarity Measure for Text Classification. In B. Benatallah, A. Bestavros, Y. Manolopoulos, A. Vakali, & Y. Zhang (Eds.), *Web Information Systems Engineering –*

WISE 2014 (Vol. 8786, pp. 105–114). Springer International Publishing. https://doi.org/10.1007/978-3-319-11749-2_8

Arora, S., Narayan, A., Chen, M. F., Orr, L., Guha, N., Bhatia, K., Chami, I., Sala, F., & Ré, C. (2022). *Ask Me Anything: A simple strategy for prompting language models* (No. arXiv:2210.02441). arXiv. https://doi.org/10.48550/arXiv.2210.02441

Bastan, M., Surdeanu, M., & Balasubramanian, N. (2023). NEUROSTRUCTURAL DECODING: Neural Text Generation with Structural Constraints. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 9496–9510. https://doi.org/10.18653/v1/2023.acl(long).528

Bernhard, D., Ligozat, A.-L., Martin, F., Bras, M., Magistry, P., Vergez-Couret, M., Steiblé, L., Erhart, P., Hathout, N., Huck, D., Rey, C., Reynés, P., Rosset, S., Sibille, J., & Lavergne, T. (2018). Corpora with part-of-speech annotations for three regional languages of France: Alsatian, Occitan and Picard. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (pp. 1367–1372). Miyazaki, Japan: European Language Resources Association (ELRA). https://aclanthology.org/L18-1367/

Blevins, T., Gonen, H., & Zettlemoyer, L. (2023). *Prompting Language Models for Linguistic Structure* (No. arXiv:2211.07830). arXiv. https://doi.org/10.48550/arXiv.2211.07830

Bohnet, B. (2016). *Comparing Advanced Graph-based and Transition-based Dependency Parsers*.

Bortz, J., & Schuster, C. (2011). *Statistik für Human- und Sozialwissenschaftler: Limitierte Sonderausgabe* (7. Aufl. 2010). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-12770-0

Bras, M., & Vergez-Couret, M. (2016). BaTelÒc: A text base for the Occitan language. In V. Ferreira & P. Bouda (Eds.), *Language documentation and conservation in Europe* (pp. 133–149). University of Hawai'i Press.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners* (No. arXiv:2005.14165). arXiv. http://arxiv.org/abs/2005.14165

Calin, W. (2001). *Minority Literatures and Modernism: Scots, Breton, and Occitan, 1920-1990*. University of Toronto Press. https://doi.org/10.3138/9781442677289

Chen, X., & Wan, X. (2024). *Evaluating, Understanding, and Improving Constrained Text Generation for Large Language Models* (No. arXiv:2310.16343). arXiv. https://doi.org/10.48550/arXiv.2310.16343

Cohen, J. (2013). *Statistical Power Analysis for the Behavioral Sciences* (0 ed.). Routledge. https://doi.org/10.4324/9780203771587

Dang, H., Mecke, L., Lehmann, F., Goller, S., & Buschek, D. (2022). *How to Prompt? Opportunities and Challenges of Zero- and Few-Shot Learning for Human-AI Interaction in Creative Applications of Generative Models* (No. arXiv:2209.01390). arXiv. https://doi.org/10.48550/arXiv.2209.01390

Dang, J., Ahmadian, A., Marchisio, K., Kreutzer, J., Üstün, A., & Hooker, S. (2024). RLHF Can Speak Many Languages: Unlocking Multilingual Preference Optimization for LLMs. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 13134–13156. https://doi.org/10.18653/v1/2024.emnlp-main.729

Dang, J., Singh, S., D'souza, D., Ahmadian, A., Salamanca, A., Smith, M., Peppin, A., Hong, S., Govindassamy, M., Zhao, T., Kublik, S., Amer, M., Aryabumi, V., Campos, J. A., Tan, Y.-C., Kocmi, T., Strub, F., Grinsztajn, N., Flet-Berliac, Y., … Hooker, S. (2024). *Aya Expanse: Combining Research Breakthroughs for a New Multilingual Frontier* (No. arXiv:2412.04261). arXiv. https://doi.org/10.48550/arXiv.2412.04261

Davidson, J. (2022). On Catalan as a minority language: The case of Catalan laterals in Barcelonan Spanish. *Journal of Sociolinguistics*, *26*(3), 362–385. https://doi.org/10.1111/josl.12545

De Marneffe, M.-C., Manning, C. D., Nivre, J., & Zeman, D. (2021). Universal Dependencies. *Computational Linguistics*, 1–54. https://doi.org/10.1162/coli_a_00402

de-Dios-Flores, I., Suárez, S. P., Pérez, C. C., Outeiriño, D. B., Garcia, M., & Gamallo, P. (2024). CorpusNÓS: A massive Galician corpus for training large language models. In P. Gamallo, D. Claro, A. Teixeira, L. Real, M. Garcia, H. G. Oliveira, & R. Amaro (Eds.), *Proceedings of the 16th International Conference on Computational Processing of Portuguese—Vol. 1* (pp. 593–599). Association for Computational Lingustics. https://aclanthology.org/2024.propor-1.66/

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (No. arXiv:1810.04805). arXiv. https://doi.org/10.48550/arXiv.1810.04805

Eberhard, D. M., Simons, G. F., & Fennig, C. D. (Eds.). (2023). *Ethnologue: Languages of the World* (26th ed.). SIL International. https://www.ethnologue.com/

Fahrmeir, L., Heumann, C., Künstler, R., Pigeot, I., & Tutz, G. (2023). *Statistik: Der Weg zur Datenanalyse*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-67526-7

Geng, S., Döner, B., Wendler, C., Josifoski, M., & West, R. (2024). *Sketch-Guided Constrained Decoding for Boosting Blackbox Large Language Models without Logit Access* (No. arXiv:2401.09967). arXiv. https://doi.org/10.48550/arXiv.2401.09967

Gonzalez-Agirre, A., Marimon, M., Rodriguez-Penagos, C., Aula-Blasco, J., Baucells, I., Armentano-Oller, C., Palomar-Giner, J., Kulebi, B., & Villegas, M. (2024). Building a Data Infrastructure for a Mid-Resource Language: The Case of Catalan. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, & N. Xue (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and*

Evaluation (LREC-COLING 2024) (pp. 2556–2566). ELRA and ICCL. https://aclanthology.org/2024.lrec-main.231/

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., … Ma, Z. (2024). *The Llama 3 Herd of Models* (No. arXiv:2407.21783). arXiv. https://doi.org/10.48550/arXiv.2407.21783

Grevisse, M., & Goosse, A. (2016). *Le Bon Usage* (16th ed.). De Boeck Supérieur.

Hammarström, H., Forkel, R., Haspelmath, M., & Bank, S. (2025). *glottolog: Glottolog database 5.2.1* (Version v5.2.1) [Dataset]. Zenodo. https://doi.org/10.5281/ZENODO.15640155

Howard, J., & Ruder, S. (2018). *Universal Language Model Fine-tuning for Text Classification* (No. arXiv:1801.06146). arXiv. https://doi.org/10.48550/arXiv.1801.06146

Hromei, C. D., Croce, D., & Basili, R. (2024). U-DepPLLaMA: Universal Dependency Parsing via Auto-regressive Large Language Models. *Italian Journal of Computational Linguistics*, *10*(1). https://doi.org/10.4000/125nm

Hualde, J. I., & Ortiz De Urbina, J. (Eds.). (2003). *A Grammar of Basque:* DE GRUYTER MOUTON. https://doi.org/10.1515/9783110895285

Jaccard, P. (1912). THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.[1]. *New Phytologist*, *11*(2), 37–50. https://doi.org/10.1111/j.1469-8137.1912.tb05611.x

Joshi, P., Santy, S., Budhiraja, A., Bali, K., & Choudhury, M. (2020). The State and Fate of Linguistic Diversity and Inclusion in the NLP World. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6282–6293. https://doi.org/10.18653/v1/2020.acl-main.560

Jurafsky, D., & Martin, J. H. (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models* (3rd ed.). Stanford University. https://web.stanford.edu/~jurafsky/slp3

Khandelwal, K., Tonneau, M., Bean, A. M., Kirk, H. R., & Hale, S. A. (2024). Indian-BhED: A Dataset for Measuring India-Centric Biases in Large Language Models. *Proceedings of the 2024 International Conference on Information Technology for Social Good*, 231–239. https://doi.org/10.1145/3677525.3678666

Khondaker, M. T. I., Waheed, A., Nagoudi, E. M. B., & Abdul-Mageed, M. (2023). *GPTAraEval: A Comprehensive Evaluation of ChatGPT on Arabic NLP* (No. arXiv:2305.14976). arXiv. https://doi.org/10.48550/arXiv.2305.14976

Kim, K., & Ko, Y. (2025). *Dependency Parsing with the Structuralized Prompt Template* (No. arXiv:2502.16919). arXiv. https://doi.org/10.48550/arXiv.2502.16919

Kumawat, D., & Jain, V. (2015). POS Tagging Approaches: A Comparison. *International Journal of Computer Applications*, *118*(6), 32–38. https://doi.org/10.5120/20752-3148

Kunchukuttan, A., Jain, S., & Kejriwal, R. (2021). A Large-scale Evaluation of Neural Machine Transliteration for Indic Languages. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 3469–3475. https://doi.org/10.18653/v1/2021.eacl-main.303

Li, T., Chiang, W.-L., Frick, E., Dunlap, L., Wu, T., Zhu, B., Gonzalez, J. E., & Stoica, I. (2024). *From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline* (No. arXiv:2406.11939). arXiv. https://doi.org/10.48550/arXiv.2406.11939

Lin, B., Yao, Z., Shi, J., Cao, S., Tang, B., Li, S., Luo, Y., Li, J., & Hou, L. (2022). Dependency Parsing via Sequence Generation. *Findings of the Association for Computational*

Linguistics: EMNLP 2022, 7339–7353. https://doi.org/10.18653/v1/2022.findings-emnlp.543

Lin, B., Zhou, X., Tang, B., Gong, X., & Li, S. (2023). *ChatGPT is a Potential Zero-Shot Dependency Parser* (No. arXiv:2310.16654). arXiv. https://doi.org/10.48550/arXiv.2310.16654

Littell, P., Mortensen, D. R., Lin, K., Kairis, K., Turner, C., & Levin, L. (2017). URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 8–14. https://doi.org/10.18653/v1/E17-2002

Liu, H. (2008). Dependency Distance as a Metric of Language Comprehension Difficulty. *Journal of Cognitive Science*, *9*(2), 159–191. https://doi.org/10.17791/JCS.2008.9.2.159

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach* (No. arXiv:1907.11692). arXiv. https://doi.org/10.48550/arXiv.1907.11692

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60. https://doi.org/10.3115/v1/P14-5010

Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2024). Prompt Engineering in Large Language Models. In I. J. Jacob, S. Piramuthu, & P. Falkowski-Gilski (Eds.), *Data Intelligence and Cognitive Informatics* (pp. 387–402). Springer Nature Singapore. https://doi.org/10.1007/978-981-99-7962-2_30

Dryer, M., & Haspelmath, M. (2024). *The World Atlas of Language Structures Online* (Version v2020.4) [Dataset]. Zenodo. https://doi.org/10.5281/ZENODO.13950591

Nivre, J. (2003). An efficient algorithm for projective dependency parsing. *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT 2003)*, 149–160. https://aclanthology.org/W03-0507/

Nivre, J. (2020). Universal Dependencies v2: An evergrowing multilingual treebank collection. *Language Resources and Evaluation*, 54(4), 1005–1020. https://doi.org/10.1007/s10579-020-09476-2

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., & Zeman, D. (2016). Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)* (pp. 1659–1666). Portorož, Slovenia: European Language Resources Association (ELRA). https://aclanthology.org/L16-1262.pdf

Odumakinde, A., D'souza, D., Verga, P., Ermis, B., & Hooker, S. (2024). *Multilingual Arbitrage: Optimizing Data Pools to Accelerate Multilingual Progress* (No. arXiv:2408.14960). arXiv. https://doi.org/10.48550/arXiv.2408.14960

Pearson, K. (1896). VII. Mathematical contributions to the theory of evolution.—III. Regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, *187*, 253–318. https://doi.org/10.1098/rsta.1896.0007

Peeperkorn, M., Kouwenhoven, T., Brown, D., & Jordanous, A. (2024). *Is Temperature the Creativity Parameter of Large Language Models?* (No. arXiv:2405.00492). arXiv. https://doi.org/10.48550/arXiv.2405.00492

Pöckl, W., Rainer, F., & Pöll, B. (2013). *Einführung in die romanische Sprachwissenschaft* (5th rev. ed.). Walter de Gruyter.

Poujade, C., Bras, M., & Urieli, A. (2024). CorpusArièja: Building an Annotated Corpus with Variation in Occitan. In M. Melero, S. Sakti, & C. Soria (Eds.), *Proceedings of the 3rd*

Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024 (pp. 66–71). ELRA and ICCL. https://aclanthology.org/2024.sigul-1.9/

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). *Stanza: A Python Natural Language Processing Toolkit for Many Human Languages* (No. arXiv:2003.07082). arXiv. https://doi.org/10.48550/arXiv.2003.07082

Qwen, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., … Qiu, Z. (2025). *Qwen2.5 Technical Report* (No. arXiv:2412.15115). arXiv. https://doi.org/10.48550/arXiv.2412.15115

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training*. OpenAI. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners*. OpenAI. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Ramirez-Mitjans, S., Ni, H., Moseguí, J., Puerta, J., Vera, M., & Gibert, K. (2024). Pushing the Boundaries of Natural Language Processing (NLP): Enhancing Catalan Text Transcription Through Large-Scale Models in the Educational Field. In T. Alsinet, X. Vilasís, D. García, & E. Álvarez (Eds.), *Frontiers in Artificial Intelligence and Applications*. IOS Press. https://doi.org/10.3233/FAIA240428

Raspanti, F., Ozcelebi, T., & Holenderski, M. (2025). Grammar-Constrained Decoding Makes Large Language Models Better Logical Parsers. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, 485–499. https://doi.org/10.18653/v1/2025.acl-industry.34

Ratcliff, J. W., & Metzener, D. E. (1988). Pattern Matching: The Gestalt Approach. *Dr. Dobb's Journal*, *13*(7), 46–72.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 1996)* (pp. 133–142). Philadelphia, PA: Association for Computational Linguistics. https://aclanthology.org/W96-0213/

Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R., & Gal, Y. (2024). AI models collapse when trained on recursively generated data. *Nature*, *631*(8022), 755–759. https://doi.org/10.1038/s41586-024-07566-y

Singh, S., Vargus, F., Dsouza, D., Karlsson, B. F., Mahendiran, A., Ko, W.-Y., Shandilya, H., Patel, J., Mataciunas, D., OMahony, L., Zhang, M., Hettiarachchi, R., Wilson, J., Machado, M., Moura, L. S., Krzemiński, D., Fadaei, H., Ergün, I., Okoh, I., … Hooker, S. (2024). *Aya Dataset: An Open-Access Collection for Multilingual Instruction Tuning* (No. arXiv:2402.06619). arXiv. https://doi.org/10.48550/arXiv.2402.06619

Sousa, X. (2020). From Regional Dialects to the Standard: Measuring Linguistic Distance in Galician Varieties. *Languages*, *5*(1), 4. https://doi.org/10.3390/languages5010004

Spearman, C. (1904). The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, *15*(1), 72. https://doi.org/10.2307/1412159

Straka, M. (2018). UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In D. Zeman & J. Hajič (Eds.), *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (pp. 197–207). Association for Computational Linguistics. https://doi.org/10.18653/v1/K18-2020

Taulé, M., Martí, M. A., & Recasens, M. (2008). AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, & D. Tapias (Eds.), *Proceedings of the Sixth International Conference on*

Language Resources and Evaluation (LREC'08). European Language Resources Association (ELRA). https://aclanthology.org/L08-1222/

Team, G., Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., Ferret, J., Liu, P., Tafti, P., Friesen, A., Casbon, M., Ramos, S., Kumar, R., Lan, C. L., Jerome, S., … Andreev, A. (2024). *Gemma 2: Improving Open Language Models at a Practical Size* (No. arXiv:2408.00118). arXiv. https://doi.org/10.48550/arXiv.2408.00118

Tesnière, L. (1959). *Éléments de syntaxe structurale*. Klincksieck.

Trautmann, D., Petrova, A., & Schilder, F. (2022). *Legal Prompt Engineering for Multilingual Legal Judgement Prediction* (No. arXiv:2212.02199). arXiv. https://doi.org/10.48550/arXiv.2212.02199

Urieli, A. (2013). *Robust French syntax analysis: Reconciling statistical methods and linguistic knowledge in the Talismane toolkit* (Doctoral dissertation). Université Toulouse II – Le Mirail. https://www.researchgate.net/publication/278644983_Robust_French_syntax_analysis_reconciling_statistical_methods_and_linguistic_knowledge_in_the_Talismane_toolkit

Vashishtha, A., Ahuja, K., & Sitaram, S. (2023). *On Evaluating and Mitigating Gender Biases in Multilingual Settings* (No. arXiv:2307.01503). arXiv. https://doi.org/10.48550/arXiv.2307.01503

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT* (No. arXiv:2302.11382). arXiv. https://doi.org/10.48550/arXiv.2302.11382

Winata, G. I., Madotto, A., Lin, Z., Liu, R., Yosinski, J., & Fung, P. (2021). Language Models are Few-shot Multilingual Learners. *Proceedings of the 1st Workshop on Multilingual Representation Learning*, 1–15. https://doi.org/10.18653/v1/2021.mrl-1.1

Yadav, P., Vu, T., Lai, J., Chronopoulou, A., Faruqui, M., Bansal, M., & Munkhdalai, T. (2024). *What Matters for Model Merging at Scale?* (No. arXiv:2410.03617). arXiv. https://doi.org/10.48550/arXiv.2410.03617

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2020). *XLNet: Generalized Autoregressive Pretraining for Language Understanding* (No. arXiv:1906.08237). arXiv. https://doi.org/10.48550/arXiv.1906.08237

Yong, Z.-X., Menghini, C., & Bach, S. H. (2024). *Low-Resource Languages Jailbreak GPT-4* (No. arXiv:2310.02446). arXiv. https://doi.org/10.48550/arXiv.2310.02446

Yoo, A. B., Jette, M. A., & Grondona, M. (2003). SLURM: Simple Linux Utility for Resource Management. In D. Feitelson, L. Rudolph, & U. Schwiegelshohn (Eds.), *Job Scheduling Strategies for Parallel Processing* (Vol. 2862, pp. 44–60). Springer Berlin Heidelberg. https://doi.org/10.1007/10968987_3

Zeman, D., & Resnik, P. (2008). Cross-Language Parser Adaptation between Related Languages. *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*. https://aclanthology.org/I08-3008/

Zhang, S., Bao, Y., & Huang, S. (2024). *EDT: Improving Large Language Models' Generation by Entropy-based Dynamic Temperature Sampling* (No. arXiv:2403.14541). arXiv. https://doi.org/10.48550/arXiv.2403.14541

Zhang, Y., & Clark, S. (2008). A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing using beam-search. *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*, 562. https://doi.org/10.3115/1613715.1613784

Zhao, T. Z., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021). *Calibrate Before Use: Improving Few-Shot Performance of Language Models* (No. arXiv:2102.09690). arXiv. https://doi.org/10.48550/arXiv.2102.09690

Appendix

Code and data: https://github.com/maralb99/syntax-parsing-icl-romance-langs.git

In-Context Learning Results for minority languages (ca, gl, oc) for unconstrained and constrained decoding

	Unconstrained Decoding			Constrained Decoding			Structural Parser		
	UPOS	UAS	LAS	UPOS	UAS	LAS	UPOS	UAS	LAS
ca_ca									
0-shot	69.21	28.39	21.86	92.36	44.34	36.31	92.36	22.70	17.12
1-shot (short)	77.13	33.63	26.94	92.64	44.23	36.14	-	-	-
1-shot	-	-	-	93.70	45.01	37.87	-	-	-
1-shot (long)	77.58	40.10	32.18	93.70	46.12	39.43	-	-	-
3-shot	62.41	38.82	31.90	94.70	45.34	38.82	-	-	-
5-shot	68.27	35.97	28.67	94.53	45.34	38.93	-	-	-
10-shot	69.10	37.42	31.29	95.59	45.51	39.04	-	-	-
ca_es									
0-shot	69.21	28.39	21.86	92.36	44.34	36.31	92.36	22.70	17.12
1-shot (short)	82.38	45.18	35.64	92.08	43.73	37.03	-	-	-
1-shot	-	-	-	91.19	43.39	37.98	-	-	-
1-shot (long)	84.38	52.09	41.61	92.69	44.73	38.43	-	-	-
3-shot	66.48	41.49	34.80	92.47	44.73	38.65	-	-	-
5-shot	72.95	43.22	35.30	92.69	45.06	39.04	-	-	-
10-shot	76.63	45.23	37.70	92.58	43.39	37.48	-	-	-
ca_oc									
0-shot	69.21	28.39	21.86	92.36	44.34	36.31	92.36	22.70	17.12
1-shot (short)	83.38	45.34	35.14	92.19	44.56	37.09	-	-	-
1-shot	-	-	-	93.03	46.63	38.82	-	-	-
1-shot (long)	83.32	47.74	36.87	92.53	44.56	38.04	-	-	-
3-shot	78.47	45.45	37.53	92.02	44.95	38.43	-	-	-
5-shot	79.42	47.18	37.76	91.52	44.79	38.43	-	-	-
10-shot	86.73	48.35	39.88	92.25	45.45	39.10	-	-	-
ca_mixed									
0-shot	69.21	28.39	21.86	92.36	44.34	36.31	92.36	22.70	17.12
3-shot	72.39	39.21	30.23	91.91	45.06	38.32	-	-	-
5-shot	66.93	38.71	30.90	92.69	44.95	38.43	-	-	-
10-shot	84.77	45.79	35.86	92.81	46.18	39.60	-	-	-

Table A.1: all results for Catalan for unconstrained and constrained decoding; scores in %

	Unconstrained Decoding			Constrained Decoding			Structural Parser		
	UPOS	UAS	LAS	UPOS	UAS	LAS	UPOS	UAS	LAS
gl_gl									
0-shot	82.44	33.15	23.92	91.22	44.79	38.55	91.22	23.64	16.38
1-shot (short)	69.61	38.89	26.67	91.16	43.95	37.82	-	-	-
1-shot	-	-	-	91.95	44.57	39.67	-	-	-
1-shot (long)	73.38	39.84	30.61	92.07	46.31	41.31	-	-	-
3-shot	70.51	33.82	25.94	92.40	45.92	41.70	-	-	-
5-shot	72.14	35.57	27.86	92.35	45.69	41.31	-	-	-
10-shot	82.16	37.93	31.80	93.70	44.46	40.29	-	-	-
gl_es									
0-shot	82.44	33.15	23.92	91.22	44.79	38.55	91.22	23.64	16.38
1-shot (short)	70.34	39.17	27.91	90.77	44.18	38.83	-	-	-
1-shot	-	-	-	90.77	44.74	39.05	-	-	-
1-shot (long)	83.74	46.71	34.16	91.28	45.69	40.12	-	-	-
3-shot	57.46	34.16	23.30	90.71	45.41	40.41	-	-	-
5-shot	55.71	29.38	20.15	90.83	44.85	39.62	-	-	-
10-shot	66.35	33.76	22.28	90.26	43.84	39.00	-	-	-
gl_pt									
0-shot	82.44	33.15	23.92	91.22	44.79	38.55	91.22	23.64	16.38
1-shot (short)	74.96	34.50	24.03	91.05	45.58	38.89	-	-	-
1-shot	-	-	-	91.90	46.15	40.52	-	-	-
1-shot (long)	73.89	40.63	29.09	92.01	45.86	39.90	-	-	-
3-shot	75.13	37.48	27.01	92.12	46.48	41.19	-	-	-
5-shot	58.75	33.03	22.51	91.95	46.43	41.59	-	-	-
10-shot	66.80	36.69	27.29	92.40	46.15	40.97	-	-	-
gl_mixed									
0-shot	82.44	33.15	23.92	91.22	44.79	38.55	91.22	23.64	16.38
3-shot	61.45	34.27	23.24	91.90	45.81	40.29	-	-	-
5-shot	67.30	38.21	27.52	92.07	46.54	41.19	-	-	-
10-shot	75.46	34.27	25.55	92.12	45.86	41.02	-	-	-

Table A.2: all results for Galician for unconstrained and constrained decoding; scores in %

	Unconstrained Decoding			Constrained Decoding			Structural Parser		
	UPOS	UAS	LAS	UPOS	UAS	LAS	UPOS	UAS	LAS
oc_oc									
0-shot	74.64	35.32	24.88	85.47	44.48	33.67	85.47	29.62	19.59
1-shot (short)	78.52	46.60	32.65	88.29	45.38	36.49	-	-	-
1-shot	-	-	-	89.08	46.51	36.71	-	-	-
1-shot (long)	85.92	49.27	39.81	89.64	46.40	37.05	-	-	-
3-shot	80.83	43.69	34.59	88.63	44.93	35.36	-	-	-
5-shot	74.51	50.24	42.96	88.85	45.38	36.04	-	-	-
10-shot	83.98	45.39	41.26	89.08	47.07	38.18	-	-	-
oc_fr									
0-shot	74.64	35.32	24.88	85.47	44.48	33.67	85.47	29.62	19.59
1-shot (short)	80.34	44.66	33.86	87.61	46.28	35.25	-	-	-
1-shot	-	-	-	87.16	47.07	36.94	-	-	-
1-shot (long)	77.43	43.81	33.13	89.64	46.40	37.05	-	-	-
3-shot	81.92	37.99	29.13	87.61	46.62	36.60	-	-	-
5-shot	73.54	33.01	22.94	88.06	46.85	37.73	-	-	-
10-shot	75.49	37.14	27.55	88.74	46.96	36.82	-	-	-
oc_ca									
0-shot	74.64	35.32	24.88	85.47	44.48	33.67	85.47	29.62	19.59
1-shot (short)	74.03	37.86	28.52	87.61	44.93	34.23	-	-	-
1-shot	-	-	-	88.40	46.28	36.37	-	-	-
1-shot (long)	75.49	38.96	28.40	88.18	46.17	36.15	-	-	-
3-shot	56.31	35.80	26.21	87.84	45.72	35.81	-	-	-
5-shot	66.26	37.26	28.52	87.84	45.83	35.47	-	-	-
10-shot	75.61	40.41	30.95	87.50	44.26	34.12	-	-	-
oc_mixed									
0-shot	74.64	35.32	24.88	85.47	44.48	33.67	85.47	29.62	19.59
3-shot	75.36	34.83	26.94	86.26	45.72	35.47	-	-	-
5-shot	75.00	40.66	30.46	88.06	45.83	36.37	-	-	-
10-shot	77.55	37.86	28.76	88.18	47.75	37.50	-	-	-

Table A.3: all results for Occitan languages for unconstrained and constrained decoding; scores in %

Constrained Decoding Results for the target languages French and Spanish

	UPOS	UAS	LAS		UPOS	UAS	LAS
fr_fr				**es_fr**			
0-shot	95.73	48.91	43.13	0-shot	92.56	43.73	36.29
1-shot (short)	94.22	47.07	41.46	1-shot (short)	93.10	43.37	37.28
1-shot	93.80	46.57	41.79	1-shot	93.73	46.42	39.96
1-shot (long)	95.90	48.32	43.80	1-shot (long)	93.19	44.09	39.25
3-shot	95.31	48.16	42.96	3-shot	93.28	44.98	38.62
5-shot	95.56	48.58	42.88	5-shot	93.55	45.79	40.05
10-shot	95.81	49.92	44.05	10-shot	93.19	44.89	38.71
fr_es				**es_es**			
0-shot	95.73	48.91	43.13	0-shot	92.56	43.73	36.29
1-shot(short)	93.63	45.90	41.29	1-shot (short)	91.58	44.27	38.62
1-shot	93.47	47.24	42.96	1-shot	93.55	44.53	38.53
1-shot (long)	94.22	48.24	43.30	1-shot (long)	93.46	45.79	39.70
3-shot	95.31	48.66	43.38	3-shot	93.91	46.51	39.52
5-shot	94.64	47.57	42.63	5-shot	93.91	46.06	39.87
10-shot	94.39	47.82	42.80	10-shot	93.55	45.25	39.25
fr_oc				**-**			
0-shot	95.73	48.91	43.13	-	-	-	-
1-shot (short)	94.30	45.64	41.54	-	-	-	-
1-shot	95.23	49.16	43.89	-	-	-	-
1-shot (long)	93.63	47.40	42.71	-	-	-	-
3-shot	95.14	49.08	44.22	-	-	-	-
5-shot	95.14	48.16	43.72	-	-	-	-
10-shot	94.97	50.25	45.14	-	-	-	-
fr_ca				**es_ca**			
0-shot	95.73	48.91	43.13	0-shot	92.56	43.73	36.29
1-shot (short)	95.48	47.65	41.12	1-shot (short)	93.82	43.82	37.01
1-shot	94.89	49.75	44.14	1-shot	93.10	47.13	40.32
1-shot (long)	95.39	49.58	44.81	1-shot (long)	94.27	45.70	40.50
3-shot	95.31	49.58	44.39	3-shot	93.10	45.16	38.53
5-shot	94.97	49.33	44.30	5-shot	93.01	44.89	38.44
10-shot	95.56	48.74	43.38	10-shot	93.55	45.61	39.16
fr_mixed				**es_mixed**			
0-shot	95.73	48.91	43.13	0-shot	92.56	43.73	36.29
3-shot	95.39	47.82	43.05	3-shot	94.35	46.15	39.78
5-shot	95.14	48.83	43.47	5-shot	94.09	44.98	39.43
10-shot	95.23	50.17	44.81	10-shot	93.55	46.06	39.43

Table A.4: all constrained decoding results for target languages French and Spanish; scores in %

Constrained Decoding Results for the target languages Portuguese and Basque

	UPOS	UAS	LAS		UPOS	UAS	LAS
pt_fr				**eu_fr**			
0-shot	91.52	49.03	41.44	0-shot	76.29	35.05	15.46
1-shot (short)	91.60	46.20	41.20	1-shot (short)	75.60	35.57	17.35
1-shot	92.08	47.66	41.60	1-shot	73.88	34.36	15.81
1-shot (long)	92.08	47.58	42.25	1-shot (long)	76.63	38.83	17.70
3-shot	92.25	49.03	43.46	3-shot	75.77	37.46	16.84
5-shot	92.16	47.66	42.33	5-shot	74.23	37.11	17.35
10-shot	91.60	46.20	40.95	10-shot	77.49	39.00	18.21
pt_es				**eu_es**			
0-shot	91.52	49.03	41.44	0-shot	76.29	35.05	15.46
1-shot (short)	91.28	44.67	40.23	1-shot (short)	72.16	34.36	15.81
1-shot	91.44	46.12	41.76	1-shot	74.57	39.00	18.38
1-shot (long)	92.08	46.93	42.65	1-shot (long)	76.29	39.00	19.07
3-shot	91.92	48.30	43.21	3-shot	78.35	37.80	18.21
5-shot	91.11	46.04	42.00	5-shot	78.52	39.35	18.73
10-shot	91.28	46.28	41.36	10-shot	80.07	37.80	17.70
pt_pt				**eu_eu**			
0-shot	91.52	49.03	41.44	0-shot	76.29	35.05	15.46
1-shot (short)	92.08	47.01	41.28	1-shot (short)	76.98	39.00	17.70
1-shot	92.00	48.47	43.54	1-shot	78.35	36.94	18.73
1-shot (long)	93.13	47.98	42.57	1-shot (long)	84.36	43.99	21.13
3-shot	92.89	48.87	43.94	3-shot	82.99	41.58	19.76
5-shot	92.33	47.74	42.97	5-shot	84.36	41.24	21.31
10-shot	93.78	49.11	44.02	10-shot	85.57	41.75	21.31
pt_gl				**eu_oc**			
0-shot	91.52	49.03	41.44	0-shot	76.29	35.05	15.46
1-shot (short)	92.08	47.74	42.00	1-shot (short)	76.29	35.57	17.01
1-shot	92.25	47.25	42.08	1-shot	77.15	40.38	19.24
1-shot (long)	92.00	49.27	43.54	1-shot (long)	75.95	35.91	16.49
3-shot	91.76	47.66	43.54	3-shot	77.15	38.49	20.96
5-shot	92.33	47.33	42.73	5-shot	75.95	37.80	19.24
10-shot	92.41	46.45	41.68	10-shot	73.37	35.74	17.53
pt_mixed				**eu_mixed**			
0-shot	91.52	49.03	41.44	0-shot	76.29	35.05	15.46
3-shot	92.00	48.71	43.70	3-shot	76.63	37.46	18.38
5-shot	91.92	47.98	43.38	5-shot	78.52	37.63	17.70
10-shot	92.89	46.61	42.25	10-shot	77.84	38.14	18.73

Table A.5: all constrained decoding results for target languages Portuguese and Basque; scores in %

Significance test results
only statistically significant results (p-value ≤ 0.05)
Comparison type: shot setting
effect size: s = small, n = negligible; setting = shot setting; ES = effect size
p_value is rounded to 6 decimals and diff to 2 decimals

lang_pair	metric	setting1	setting2	score1	score2	diff	p_value	ES
eu_eu	UPOS	0-shot	10-shot	76.29	85.57	9.28	0.000056	s
eu_eu	UPOS	1-shot(short)	10-shot	76.98	85.57	8.59	0.000173	s
eu_eu	UPOS	0-shot	1-shot(long)	76.29	84.36	8.07	0.000534	s
eu_eu	UPOS	0-shot	5-shot	76.29	84.36	8.07	0.000534	s
eu_es	UPOS	1-shot(short)	10-shot	72.16	80.07	7.91	0.001553	n
eu_eu	UPOS	1-shot(short)	1-shot(long)	76.98	84.36	7.38	0.001432	n
eu_eu	UPOS	1-shot(short)	5-shot	76.98	84.36	7.38	0.001432	n
eu_eu	UPOS	1-shot	10-shot	78.35	85.57	7.22	0.001360	n
eu_eu	UPOS	0-shot	3-shot	76.29	82.99	6.7	0.004535	n
eu_es	UPOS	1-shot(short)	5-shot	72.16	78.52	6.36	0.011834	n
eu_es	UPOS	1-shot(short)	3-shot	72.16	78.35	6.19	0.014407	n
eu_eu	UPOS	1-shot	1-shot(long)	78.35	84.36	6.01	0.008479	n
eu_eu	UPOS	1-shot	5-shot	78.35	84.36	6.01	0.008479	n
eu_eu	UPOS	1-shot(short)	3-shot	76.98	82.99	6.01	0.010397	n
eu_es	UPOS	1-shot	10-shot	74.57	80.07	5.5	0.025060	n
eu_eu	UPOS	1-shot	3-shot	78.35	82.99	4.64	0.045023	n
oc_oc	UPOS	0-shot	1-shot(long)	85.47	89.64	4.17	0.010342	n
oc_fr	UPOS	0-shot	1-shot(long)	85.47	89.64	4.17	0.010342	n
oc_oc	UPOS	0-shot	10-shot	85.47	89.08	3.61	0.027893	n
oc_oc	UPOS	0-shot	1-shot	85.47	89.08	3.61	0.027893	n
oc_oc	UPOS	0-shot	5-shot	85.47	88.85	3.38	0.040286	n
oc_fr	UPOS	0-shot	10-shot	85.47	88.74	3.27	0.047653	n
ca_ca	UPOS	0-shot	10-shot	92.36	95.59	3.23	0.000048	n
ca_ca	UPOS	1-shot(short)	10-shot	92.64	95.59	2.95	0.000175	n
gl_gl	UPOS	1-shot(short)	10-shot	91.16	93.7	2.54	0.004206	n
gl_gl	UPOS	0-shot	10-shot	91.22	93.7	2.48	0.005114	n
ca_ca	UPOS	0-shot	3-shot	92.36	94.7	2.34	0.004397	n
es_es	UPOS	1-shot(short)	3-shot	91.58	93.91	2.33	0.033853	n
es_es	UPOS	1-shot(short)	5-shot	91.58	93.91	2.33	0.033853	n
pt_pt	UPOS	0-shot	10-shot	91.52	93.78	2.26	0.031185	n
ca_ca	UPOS	0-shot	5-shot	92.36	94.53	2.17	0.008659	n
fr_fr	UPOS	1-shot	1-shot(long)	93.8	95.9	2.1	0.020255	n
ca_ca	UPOS	1-shot(short)	3-shot	92.64	94.7	2.06	0.011308	n
fr_fr	UPOS	1-shot	10-shot	93.8	95.81	2.01	0.026900	n
ca_ca	UPOS	1-shot	10-shot	93.7	95.59	1.89	0.011948	n
ca_ca	UPOS	1-shot(long)	10-shot	93.7	95.59	1.89	0.011948	n

Table A 6: statistically significant shot-setting results

lang_pair	metric	setting1	setting2	score1	score2	diff	p_value	ES
ca_ca	UPOS	1-shot(short)	5-shot	92.64	94.53	1.89	0.020911	n
gl_gl	UPOS	1-shot	10-shot	91.95	93.7	1.75	0.043252	n
fr_es	UPOS	0-shot	1-shot	95.73	93.47	-2.26	0.014559	n
fr_es	UPOS	0-shot	1-shot(short)	95.73	93.63	-2.1	0.022240	n
fr_oc	UPOS	0-shot	1-shot(long)	95.73	93.63	-2.1	0.022240	n
fr_fr	UPOS	0-shot	1-shot	95.73	93.8	-1.93	0.034243	n
eu_eu	UAS	0-shot	1-shot(long)	35.05	43.99	8.94	0.001812	n
eu_eu	UAS	1-shot	1-shot(long)	36.94	43.99	7.05	0.014276	n
eu_eu	UAS	0-shot	10-shot	35.05	41.75	6.7	0.018774	n
eu_eu	UAS	0-shot	3-shot	35.05	41.58	6.53	0.021945	n
eu_eu	UAS	0-shot	5-shot	35.05	41.24	6.19	0.029716	n
fr_oc	UAS	1-shot(short)	10-shot	45.64	50.25	4.61	0.024153	n
pt_es	UAS	0-shot	1-shot(short)	49.03	44.67	-4.36	0.029718	n
eu_eu	LAS	0-shot	10-shot	15.46	21.31	5.85	0.009988	n
eu_eu	LAS	0-shot	5-shot	15.46	21.31	5.85	0.009988	n
eu_eu	LAS	0-shot	1-shot(long)	15.46	21.13	5.67	0.012359	n
eu_oc	LAS	0-shot	3-shot	15.46	20.96	5.5	0.015053	n
es_ca	LAS	0-shot	1-shot(long)	36.29	40.5	4.21	0.040873	n
gl_gl	LAS	1-shot(short)	3-shot	37.82	41.7	3.88	0.018119	n
gl_gl	LAS	1-shot(short)	5-shot	37.82	41.31	3.49	0.033385	n
gl_gl	LAS	1-shot(short)	1-shot(long)	37.82	41.31	3.49	0.033385	n
ca_ca	LAS	1-shot(short)	1-shot(long)	36.14	39.43	3.29	0.042182	n
ca_mixed	LAS	0-shot	10-shot	36.31	39.6	3.29	0.042363	n

Table A.6 (continued): statistically significant shot-setting results

Significance test results
Comparison type: source language
effect size: s = small, n = negligible; setting = shot setting; ES = effect size
p_value is rounded to 6 decimals and diff to 2 decimals

target	source_1	source_2	#shots	metric	score1	score2	diff	p_value	ES
eu	fr	eu	5-shot	UPOS	74.23	84.36	10.13	0.000020	s
eu	fr	eu	10-shot	UPOS	77.49	85.57	8.08	0.000382	s
eu	fr	eu	3-shot	UPOS	75.77	82.99	7.22	0.002332	n
pt	fr	pt	10-shot	UPOS	91.6	93.78	2.18	0.037191	n
gl	es	pt	10-shot	UPOS	90.26	92.4	2.14	0.023398	n
ca	ca	es	5-shot	UPOS	94.53	92.69	-1.84	0.024285	n
ca	ca	mixed	5-shot	UPOS	94.53	92.69	-1.84	0.024285	n
ca	ca	es	3-shot	UPOS	94.7	92.47	-2.23	0.006429	n
pt	pt	es	10-shot	UPOS	93.78	91.28	-2.5	0.017990	n
ca	ca	oc	3-shot	UPOS	94.7	92.02	-2.68	0.001269	n
ca	ca	mixed	10-shot	UPOS	95.59	92.81	-2.78	0.000369	n
ca	ca	mixed	3-shot	UPOS	94.7	91.91	-2.79	0.000831	n
ca	ca	es	10-shot	UPOS	95.59	92.58	-3.01	0.000133	n
ca	ca	oc	5-shot	UPOS	94.53	91.52	-3.01	0.000403	n
ca	ca	oc	10-shot	UPOS	95.59	92.25	-3.34	0.000029	n
gl	gl	es	10-shot	UPOS	93.7	90.26	-3.44	0.000160	n
eu	eu	es	3-shot	UPOS	82.99	78.35	-4.64	0.045023	n
eu	eu	es	10-shot	UPOS	85.57	80.07	-5.5	0.012871	n
eu	eu	mixed	5-shot	UPOS	84.36	78.52	-5.84	0.010394	n
eu	eu	oc	3-shot	UPOS	82.99	77.15	-5.84	0.012637	n
eu	eu	es	5-shot	UPOS	84.36	78.52	-5.84	0.010394	n
eu	eu	oc	10-shot	UAS	41.75	35.74	-6.01	0.035338	n

Table A.7: statistically significant source language results

Syntactic similarity and model performance analysis

Mean performance scores for the similar group and the random group

R = mean performance score for random group; S= mean performance score in similar group; Δ = difference, similar group minus random group; rounded to 3 decimals; green highlights indicate statistically significant differences with an effect size ≥ 0.2

scenario	UPOS			LAS			UAS		
	R	S	Δ	R	S	Δ	R	S	Δ
ca_ca_1_short	0.932	0.925	**-0.006**	0.388	0.363	**-0.024**	0.459	0.481	**0.022**
ca_ca_1	0.947	0.928	**-0.019**	0.415	0.374	**-0.041**	0.501	0.453	**-0.048**
ca_ca_1_long	0.925	0.952	**0.027**	0.411	0.406	**-0.005**	0.497	0.477	**-0.02**
ca_es_1_short	0.924	0.921	**-0.004**	0.355	0.414	**0.059**	0.438	0.485	**0.047**
ca_es_1	0.933	0.894	**-0.039**	0.407	0.378	**-0.029**	0.483	0.429	**-0.054**
ca_es_1_long	0.925	0.935	**0.01**	0.395	0.395	**0**	0.481	0.456	**-0.026**
ca_oc_1_short	0.912	0.936	**0.024**	0.354	0.417	**0.063**	0.436	0.511	**0.075**
ca_oc_1	0.95	0.915	**-0.035**	0.429	0.381	**-0.047**	0.519	0.47	**-0.049**
ca_oc_1_long	0.928	0.931	**0.003**	0.404	0.382	**-0.023**	0.482	0.453	**-0.03**
es_ca_1_short	0.927	0.923	**-0.005**	0.368	0.408	**0.04**	0.45	0.493	**0.043**
es_ca_1	0.908	0.929	**0.021**	0.424	0.429	**0.004**	0.501	0.508	**0.007**
es_ca_1_long	0.902	0.956	**0.055**	0.451	0.411	**-0.039**	0.539	0.453	**-0.086**
es_es_1_short	0.885	0.923	**0.039**	0.365	0.448	**0.083**	0.425	0.516	**0.091**
es_es_1	0.904	0.939	**0.035**	0.401	0.416	**0.015**	0.459	0.495	**0.035**
es_es_1_long	0.895	0.948	**0.053**	0.446	0.413	**-0.032**	0.524	0.481	**-0.043**
es_fr_1_short	0.905	0.93	**0.025**	0.362	0.426	**0.064**	0.413	0.523	**0.11**
es_fr_1	0.901	0.94	**0.039**	0.425	0.427	**0.002**	0.494	0.51	**0.016**
es_fr_1_long	0.893	0.945	**0.051**	0.41	0.43	**0.019**	0.499	0.469	**-0.03**
eu_es_1_short	0.715	0.727	**0.012**	0.141	0.206	**0.064**	0.322	0.429	**0.107**
eu_es_1	0.739	0.733	**-0.006**	0.162	0.245	**0.082**	0.379	0.485	**0.107**
eu_es_1_long	0.75	0.75	**0.001**	0.247	0.199	**-0.048**	0.491	0.413	**-0.077**
eu_eu_1_short	0.759	0.801	**0.042**	0.2	0.211	**0.011**	0.416	0.455	**0.038**
eu_eu_1	0.76	0.803	**0.042**	0.145	0.295	**0.15**	0.348	0.51	**0.163**
eu_eu_1_long	0.765	0.773	**0.008**	0.206	0.19	**-0.017**	0.465	0.406	**-0.059**
eu_fr_1_short	0.769	0.737	**-0.031**	0.182	0.202	**0.02**	0.364	0.42	**0.055**
eu_fr_1	0.727	0.734	**0.007**	0.147	0.197	**0.05**	0.342	0.402	**0.059**
eu_fr_1_long	0.801	0.736	**-0.065**	0.222	0.174	**-0.048**	0.481	0.389	**-0.092**
eu_oc_1_short	0.779	0.745	**-0.034**	0.174	0.207	**0.033**	0.36	0.432	**0.072**
eu_oc_1	0.77	0.778	**0.009**	0.172	0.258	**0.086**	0.409	0.493	**0.084**
eu_oc_1_long	0.747	0.761	**0.013**	0.207	0.162	**-0.045**	0.439	0.359	**-0.08**
fr_ca_1_short	0.976	0.918	**-0.058**	0.4	0.408	**0.009**	0.489	0.519	**0.03**
fr_ca_1	0.927	0.957	**0.031**	0.414	0.478	**0.063**	0.513	0.556	**0.042**
fr_ca_1_long	0.931	0.966	**0.035**	0.454	0.442	**-0.012**	0.568	0.496	**-0.072**
fr_es_1_short	0.93	0.922	**-0.008**	0.39	0.434	**0.044**	0.445	0.524	**0.079**
fr_es_1	0.904	0.947	**0.043**	0.384	0.471	**0.087**	0.478	0.532	**0.054**
fr_es_1_long	0.914	0.957	**0.043**	0.447	0.415	**-0.032**	0.547	0.494	**-0.052**

Table A.8: mean performance scores for the similar group and the random group

scenario	UPOS			LAS			UAS		
	R	S	Δ	R	S	Δ	R	S	Δ
fr_fr_1_short	0.947	0.923	**-0.023**	0.393	0.423	**0.031**	0.453	0.539	**0.086**
fr_fr_1	0.914	0.946	**0.032**	0.391	0.45	**0.059**	0.477	0.522	**0.045**
fr_fr_1_long	0.931	0.973	**0.042**	0.428	0.448	**0.02**	0.521	0.517	**-0.004**
fr_oc_1_short	0.923	0.952	**0.029**	0.371	0.464	**0.093**	0.442	0.529	**0.086**
fr_oc_1	0.929	0.964	**0.035**	0.401	0.473	**0.072**	0.497	0.556	**0.058**
fr_oc_1_long	0.904	0.952	**0.048**	0.452	0.411	**-0.041**	0.549	0.465	**-0.084**
gl_es_1_short	0.905	0.91	**0.005**	0.36	0.433	**0.073**	0.406	0.497	**0.09**
gl_es_1	0.902	0.914	**0.011**	0.391	0.405	**0.014**	0.445	0.469	**0.024**
gl_es_1_long	0.903	0.923	**0.02**	0.395	0.423	**0.028**	0.446	0.486	**0.04**
gl_gl_1_short	0.912	0.912	**0**	0.42	0.357	**-0.063**	0.484	0.42	**-0.064**
gl_gl_1	0.912	0.928	**0.017**	0.401	0.405	**0.004**	0.441	0.468	**0.027**
gl_gl_1_long	0.907	0.934	**0.027**	0.401	0.44	**0.039**	0.451	0.495	**0.044**
gl_pt_1_short	0.923	0.896	**-0.027**	0.392	0.399	**0.007**	0.458	0.472	**0.014**
gl_pt_1	0.911	0.928	**0.017**	0.406	0.418	**0.011**	0.462	0.477	**0.015**
gl_pt_1_long	0.936	0.903	**-0.033**	0.413	0.4	**-0.013**	0.491	0.451	**-0.04**
oc_ca_1_short	0.895	0.807	**-0.088**	0.391	0.353	**-0.038**	0.483	0.551	**0.069**
oc_ca_1	0.819	0.898	**0.08**	0.388	0.4	**0.012**	0.552	0.497	**-0.055**
oc_ca_1_long	0.819	0.904	**0.086**	0.412	0.378	**-0.034**	0.587	0.466	**-0.121**
oc_fr_1_short	0.895	0.803	**-0.092**	0.359	0.43	**0.071**	0.461	0.606	**0.146**
oc_fr_1	0.809	0.884	**0.075**	0.447	0.387	**-0.06**	0.627	0.476	**-0.152**
oc_fr_1_long	0.849	0.915	**0.067**	0.393	0.411	**0.018**	0.554	0.491	**-0.063**
oc_oc_1_short	0.861	0.884	**0.023**	0.358	0.46	**0.102**	0.485	0.567	**0.082**
oc_oc_1	0.829	0.909	**0.079**	0.409	0.388	**-0.021**	0.565	0.476	**-0.089**
oc_oc_1_long	0.849	0.915	**0.066**	0.42	0.385	**-0.034**	0.583	0.462	**-0.121**
pt_es_1_short	0.897	0.901	**0.004**	0.368	0.456	**0.089**	0.424	0.519	**0.095**
pt_es_1	0.878	0.927	**0.049**	0.409	0.435	**0.026**	0.473	0.487	**0.014**
pt_es_1_long	0.892	0.936	**0.043**	0.393	0.472	**0.079**	0.459	0.519	**0.059**
pt_fr_1_short	0.908	0.906	**-0.002**	0.412	0.437	**0.024**	0.465	0.519	**0.054**
pt_fr_1	0.89	0.936	**0.046**	0.406	0.437	**0.032**	0.515	0.495	**-0.02**
pt_fr_1_long	0.907	0.919	**0.012**	0.375	0.467	**0.093**	0.453	0.541	**0.088**
pt_gl_1_short	0.921	0.906	**-0.016**	0.419	0.434	**0.015**	0.491	0.514	**0.024**
pt_gl_1	0.884	0.935	**0.051**	0.413	0.441	**0.029**	0.492	0.498	**0.006**
pt_gl_1_long	0.893	0.935	**0.042**	0.415	0.469	**0.054**	0.511	0.517	**0.007**
pt_pt_1_short	0.93	0.901	**-0.029**	0.417	0.417	**0**	0.485	0.502	**0.017**
pt_pt_1	0.89	0.938	**0.048**	0.421	0.466	**0.044**	0.502	0.527	**0.026**
pt_pt_1_long	0.896	0.953	**0.058**	0.441	0.437	**-0.004**	0.528	0.489	**-0.038**

Table A.8: (continued): mean performance scores for the similar group and the random group