EVALUATION OF INTER-PARSER AGREEMENT

REALIZATION REPORT

Melpomeni Chioutakou Chioutakakou
Anahita Poulad

Supervised by - Yannick Parmentier
Reviewer - Sylvain Pogodalla

MSc Natural Language Processing
M1 Supervised Project

IDMC
Academic year 2020-2021

Contents

Chapter 1

Introduction

A recurring issue observed in many NLP applications, is the inability to accurately evaluate a parser's performance on un-annotated data. The need for an automated method that provides meaningful insight about the reliability of a parser in real-world unlabeled data, is the motivation behind this project.

While parsing of a text constitutes the basis of many NLP tasks, when it comes to real-world data, high-performing parsers up to this day do not appear able to efficiently execute the task. Without a gold annotation that can be used to evaluate the parser on, the assessment of a parser's performance becomes a complex task. The most common method to deal with this issue is to manually annotate a small corpus for every new domain. However, this approach proves to be time-consuming and costly since it requires human involvement and it must be performed separately for each domain. Another proposed approach is the conduction of a reliability study and this is the direction we decided to follow for the realization part of our project.

Our approach is based on the assumption that an annotation is reliable if parsers seem to agree on the dependencies assigned. If we observe similar results between different parsers, we can deduce that they have embodied a similar understanding of the annotation guidelines and we can expect that they will perform consistently under this understanding. Reliability is thus essential in order to demonstrate the validity of the parsing.

In the following chapters, we will thoroughly present our approach and our findings concerning this open research question.

All the datasets and the code used for analysis in this study are provided in a repository at *githup repository*. The authors welcome requests for additional information regarding the material presented in this report.

Chapter 2

Dependency parsing

Recently, in the scientific world of Natural Language Processing, dependency-based methods for syntactic parsing have gained great popularity. Dependency parsing is an approach that performs automatic syntactic analysis of natural languages and is related to dependency grammar. When it comes to various NLP tasks like machine translation or information extraction, dependency-based syntactic representations appear useful thanks to their transparent encoding of predicate-argument structure, while combined with machine learning they have led to the development of accurate syntactic parsers.

The two main classes of dependency parsing approaches are knowledge-based parsing and data-driven dependency parsing. Knowledge-based approaches rely on a formal description of the dependency relations within words (so-called grammar), while data-driven ones use annotated data and machine learning algorithms to parse new sentences.

Our study is focused on the data-driven dependency parsing paradigm, therefore annotated data, or else treebanks are necessary. Dependency treebanks have been created by human annotators and are parsed text corpora that annotate dependency structures.

A very important source of dependency treebanks is The Universal Dependencies project. The project of Nivre et al., [1], proves to be very important in the field of Natural language processing. It provides openly accessible and available dependency treebanks that are linguistically-oriented, computationally useful and cross-linguistically applicable.

Chapter 3

Related work

Ravi et al. [2] try to address the problem of parser accuracy estimation on various domains and topics when no gold standard parse is available. They approach this challenge by making a prediction without annotating the target text.

An easy yet inaccurate way to estimate the accuracy of the parser aside from hand annotation of the target text is to use the parser on a standard corpus(e.g. WSJ) and assume that the accuracy of the parser would be the same for the target domain. However, different studies have shown that this estimation could be incorrect. Our experiments came to validate this. The authors suggest using some characteristics of the target domain along with some belonging to the domain the parser has been trained on, in order to predict the parser's accuracy on the target domain. This way they faced a prediction task for which the inputs are the parser,its training domain, and the target domain and the desired output is the estimation of the parser's accuracy.

They start by creating some lexical features such as the length of the sentences, the number of unknown words in the target domain (that the parser has not come across in the training domain) and a language model perplexity–based feature.Using the above features and a SVM-Regression model, they observed a positive but not significant correlation between predicted accuracy and observed accuracy of the parser.

Subsequently, the authors used some parser specific features, such as the syntactic category of the root of tree outputs and the number of nodes with each label in output trees. The correlation between predicted and observed accuracy for these parser specific features was also positive and more significant compared to the lexical features.

Furthermore, the authors introduced a reference parser that is used to gauge the performance of the main parser. In other words the output trees of the reference parser are used as a gold standard parse of the text. Even though it is not necessarily a correct parse and the F-measure resulted from comparing main parser output and this gold standard is used as a feature, these agreement-based features proved capable of creating the highest correlation between predicted and observed accuracy for the parser. In our experiment, we followed the same approach and used a reference parser to generate gold parses for the un-annotated texts.

Following the feature creation phase, the authors tried to maximize the correlation by using different combinations of the features and also expand it so that both features and predictions would be made for an entire text chunk instead of one sentence.

They observed that the range of predictions from their system was smaller than the actual F-score range. Hence, even though the correlation scores were high, it did not necessarily mean that the predictions were on target. They fixed this by shifting the mean predicted F-score and skewing individual predictions based on empirical observations.

The model was able to predict the F_0 score better than the baseline guess. The final model could also predict the ranking of different parsers on out-of-domain data. In other words, the order of predicted accuracy matches the order of actual parsing accuracy results for candidate parsers.

Opitz and Frank [3] tried to predict the accuracy of an automatically generated AMR parse. Although this study is not directly related to our research question, the approach the authors chose to evaluate automatically created parses could be considered as a method to evaluate other parser outputs too.

Their suggested prediction model is domain and parser agnostic, and they do not feed any target-text related or parser-related feature to their algorithm. The authors investigated the feasibility of automatic accuracy prediction in general and the applicability of their model.

First, they evaluated their prediction model's capacity, i.e.,if the model is capable of differentiating good and bad parses. This capacity could be investigated by calculating the correlation of predicted accuracies with true accuracies on unseen data and by checking the

model's ability to assign high scores to gold parses.

The authors also suggest predicting an ensemble of metrics suitable for assessing parse quality instead of predicting just a single accuracy score. This seems like an idea worth exploring.

Furthermore, The model was also able to successfully rank different parsers using their input sentences and their output parses.

The most important information this study provided us was the understanding that it is possible to evaluate a parse without taking the characteristics of the parser and the target text into account.

Elsahar and Galle´ [4] studied how to predict the performance drop of NLP models under domain-shift, in the absence of any target domain labels. They investigate three families of metrics for measuring domain similarity : $\mathscr{H}$-divergence that reflects capacity of another classification model to distinguish between samples from source and target domains, confidence based method that concentrates on certainty of the model over its predictions and finally, reverse classification accuracy, which uses predicted values as pseudo-labels over the target domain.

The authors suggest a regression based method that can directly estimate the performance drop of a model trained on one domain and tested on another domain. This model uses simple linear regression that fits a regression line between the drop in the model accuracy and a domain-shift detection metric according to a small fixed number of labeled evaluation datasets. In the next step this regression line is used to predict the amount of performance drop.

The main idea in this study is to use similarity metrics between source and target domains to predict the performance drop of the NLP model. In our case, although we did not use metrics, we tried to analyze the difference between training data and wikipedia text samples to find patterns.

Chapter 4

Method

To analyze inter-parser agreement we had to make choices along two fundamentally orthogonal axes: data and (parsing) systems.

One of our objectives was to provide an extensive study of different parser behaviour on different data, however an exploration space of such breadth could not possibly be properly explored within the short life-time of the project. Considering the time-constraints we faced, we settled for a compromise that would provide meaning and usefulness to our results while leaving clear what the shortcomings of our approach are.

As part of this compromise, we decided to specialize on data and make the choice of the parsing system almost static. The motivation behind this design decision is twofold:

First, we consider that there is more diversity of data than there is diversity of state-of-the-art parsers, both in terms of quantity and kind. Using data as a variable allows us to set up experiments of widely different nature and find interesting contrasts in results that can be interpreted.

Secondly, switching between parsers requires substantially more effort than switching between data, as it is necessary to understand the interfaces in order to use each parser and to set up an appropriate environment to run each parser, whereas data can be handled in an almost homogeneous manner once they have been acquired (which can be a complicated task on its own). Moreover, so as to compare results from different parsers one needs to understand the inner-workings of each parser, which is a rather time-consuming task.

Finally, we needed to establish a framework to make sense of all the results we would be producing (consisting of parses of all the $(parser, dataset)$ pairs). In spite of all the methodologies that we explored during the bibliographic part of the project, we ended up adopting the standard approach which involves using the LAS, UAS and POS accuracy to compare parses. This approach has well-known limitations such as the fact that these metrics are not chance-corrected (and they are non-trivial to chance-correct), but it is also the most widely adopted approach and the metrics are simple and effective, which makes interpreting scores easier.

To interpret and compare scores, we make use of typical statistic tools. In particular we use Pearson Correlation Coefficient (Pearson's **r**) to see how correlated the agreement of pairs of parses are. This allows us to explore inter-parser agreement as means of assessing the reliability of a parser without gold-parses. It also uncovers interesting patterns in data and parser behaviour that can help understand the weaknesses and strengths of the parsers.

This was a quick overview of the applied methodology. The rest of this chapter is devoted to explaining in detail each of these components of our method. In the next section, we briefly review the parsing systems used, followed by a description of the data we used, what motivated our choices and how data acquisition was implemented when relevant. In the last section of this chapter, we explore the inter-parser agreement.

4.1 PARSERS

As mentioned earlier, we decided after consideration to not render the choice of parser a variable in our experiments. Nevertheless, we experimented with a variety of parser architectures before concluding on our final choice. More specifically, we experimented with the *CoreNLP Stanford* parser, *Spacy's* dependency parser, and the *Stanza Stanford* neuronal parser.

After conducting some experiments, the Stanza parser appeared to be the best possible option in our case. There are several reasons behind this choice:

- Stanza's neuronal parser[5] is a recognized state-of-the-art production-ready parser.
- The package provides an easy-to-use interface that abstracts all the complexity.
- The parser works with the latest UD dependency annotation specification.

- There are several pretrained parsers (with the same architecture) available for use, which conveniently, have been trained in the UD datasets we are going to use, allowing us to explore in-domain and out-of-domain performance and parser agreement.

Tables 4.1 and 4.2 show the available pretrained parsers in Stanza and their respective performance (as measured by the authors) in the test-set of the dataset that was used to train each one.

Parser train-set	POS	UAS	LAS
EWT	95.40	86.22	83.59
GUM	95.89	87.06	83.57
LinES	96.88	85.82	81.97
ParTUT	96.15	90.31	87.35

Table 4.1: English pretrained Stanza parsers

Parser train-set	POS	UAS	LAS
GSD	97.30	91.38	89.05
ParTUT	96.60	90.71	88.37
Sequoia	98.19	90.47	88.34
Spoken	95.49	75.82	70.71

Table 4.2: French pretrained Stanza parsers

Besides these 4 parsers that are each trained on a specific dataset, there is an aditional parser for the English language that is trained in all 4 datasets combined. We are also going to use this parser as it provides an interesting perspective on how domain-dependant the parsers trained in single dataset are compared to one that should generalize across more types of data.

One should expect this combined parser to outperform the other 4 parsers in every dataset except in the dataset for which the domain-specific parsers were trained. In cases where the combined parser is less performant, it would be interesting to see how large the gap is.

4.2 DATA

After deciding that data would be the center of our research, we needed to obtain a wide variety of data from different domains for the relevant part of the pipeline. This was necessary in order to ensure our exploration and findings would be relevant not only for the data we

were using, but also generalizable to other kinds of data.

Another issue to consider regarding data is whether they should be annotated or not. Annotation is an expensive process which is why most of the data are not annotated. Despite their limited availability, annotated data are still very interesting for our data-hungry exploration needs. Although we do not need gold-parses to compute agreement between parsers, these agreement scores do not give us information about the relative performance of the parsers. If gold-parses are available however, we can correlate the agreement between the parsers with the agreement with the gold-data, thus obtaining valuable insight on the extent to which inter-parser agreement can be used as a replacement for gold-data.

Fortunately for us, the Universal Dependencies (UD) project collects and archives large dependency treebanks from different domains in multiple languages that are also available to use. Given the conveniences of this resource, we will use UD as the source of our annotated data. In particular we will use all the available English and French UD treebanks. Furthermore, as mentionned in the previous section, there are also several pretrained parsers that have been trained on these datasets.

In addition to using annotated UD treebanks, we would like to extract our own corpus from the web to experiment in a realistic scenario where there are no gold-data. In this scenario conclusions will be harder to make, since results have no reference that grounds them.

For our open web data we decided to use Wikipedia as it is a common source of open data for many projects and it has a convenient API that facilitates extraction. We built two small corpora in English and in French to have compatible data to compare with the UD treebanks.

We will now go into detail about each of these sources of data and explain the kinds of data that the different UD treebanks contain, and details about the Wikipedia data extraction process.

4.2.1 UD treebanks

As mentioned beforehand we will be using all the available UD treebanks in English and French. Here is an overview of the treebanks (hyperlinked) :

4.2.1.1 English

- **ESL**
  UD English-ESL/TLE [6] contains 5124 sentences and 97681 tokens. The English as a Second Language (ESL) sentences are manually annotated with POS tags and dependency trees in the UD formalism and were randomly drawn from the Cambridge Learner Corpus First Certificate in English (FCE) corpus.

- **EWT**
  A Gold Standard Universal Dependencies Corpus for English [7] that comprises 16621 sentences, 251494 tokens and 254830 syntactic words concerning five genres of web media: weblogs, newsgroups, emails, reviews and Yahoo! answers. Trees were automatically converted into Stanford Dependencies and then hand-corrected to Universal Dependencies.

- **GUM**
  GUM (Georgetown University Multilayer corpus) [8], constitutes an open source collection of richly annotated texts from multiple text types. The corpus contains 7402 sentences, 132835 tokens and 134488 syntactic words and is collected and expanded by students as part of the curriculum in the course LING-367 “Computational Corpus Linguistics” at Georgetown University.

- **GUMReddit**
  Universal Dependencies syntax annotations from the Reddit portion of the GUM corpus. This corpus contains 895 sentences, 15923 tokens and 16286 syntactic words.

- **LinES**
  UD English-LinES [9] is the English half of the LinES Parallel Treebank with UD annotations and it contains 5243 sentences and 94217 tokens taken mostly from literature but there are also two sections with online manual data and Europarl data. The treebank is being developed continuously.

- **ParTUT**
  UD-English-ParTUT [10] is derived from a multilingual parallel treebank developed at

the University of Turin, ParTUT. It includes texts from different sources, including talks, legal texts and Wikipedia articles, among others. This corpus contains 2090 sentences, 49602 tokens and 49634 syntactic words.

- **Pronouns**
  UD English-Pronouns [11] is a dataset created in order to make pronoun identification more accurate and with a more balanced distribution across genders. The dataset is initially targeting the Independent Genitive pronouns, “hers”, (independent) “his”, (singular) “theirs”, “mine”, and (singular) “yours”. It contains 285 sentences and 1705 tokens.

- **PUD**
  The English portion of the Parallel Universal Dependencies (PUD) [12] treebanks created for the CoNLL 2017 shared task on Multilingual Parsing from Raw Text to Universal Dependencies. It contains 1000 sentences and 21176 tokens. The sentences are randomly selected from the news domain and from Wikipedia.

Table 4.3 presents a summary of the English treebanks used.

Treebank	Domain(s)	Size (Tokens)
ESL	leaner essays	97K
EWT	social networks	254K
GUM	multi	134K
GUMReddit	reddit	16K
LinES	literature, manuals and Europarl	94K
ParTUT	talks, legal texts and Wikipedia	49K
Pronouns	grammar examples	1K
PUD	news and Wikipedia	21K
Total		666K

Table 4.3: Summary of used English UD treebanks

4.2.1.2 French

- **FQB**
  The corpus UD-French-FQB is an automatic conversion [13] of the French Question-

Bank v1 [14], a corpus entirely made of questions. It contains 2289 sentences, 23349 tokens and 23901 syntactic words.

- **FTB**
  The UD-French-FTB [15], is a treebank of sentences taken from the newspaper Le Monde, which was initially manually annotated with morphological information and phrase-structure and then converted to the Universal Dependencies annotation scheme.This corpus contains 18535 sentences, 556064 tokens and 573370 syntactic words.

- **GSD**
  The UD-French-GSD[16] was converted in 2015 from the content head version of the universal dependency treebank v2.0 and has been thereafter updated independently from the previous source. It contains 16341 sentences, 389224 tokens and 400249 syntactic words.

- **ParTUT**
  UD-French-ParTUT is derived from a multilingual parallel treebank developed at the University of Turin, ParTUT [17]. It includes texts from different sources, including talks, legal texts and Wikipedia articles, among others and contains 1020 sentences, 27658 tokens and 28595 syntactic words.

- **PUD**
  The French portion of the Parallel Universal Dependencies (PUD) treebanks[18] created for the CoNLL 2017 shared task on Multilingual Parsing from Raw Text to Universal Dependencies[19]. The corpus contains 1000 sentences, 24131 tokens and 24726 syntactic words. The sentences are randomly selected from the news domain and from Wikipedia.

- **Sequoia**
  UD-French-Sequoia [16] is an automatic conversion of the Sequoia Treebank corpus French Sequoia corpus. It contains 3099 sentences, 68596 tokens and 70548 syntactic words. The original sentences of the corpus are taken from: French Europarl, French wikipedia, Newspaper Est Républicain and European Medicines Agency.

- **Spoken**
  A Universal Dependencies corpus for spoken French[20] containing 2837 sentences, 34437 tokens and 34972 syntactic words. It was converted automatically from the Rhapsodie treebank with manual corrections.

Table 4.4 presents a summary of the French treebanks used.

Treebank	Domain(s)	Size (Tokens)
FQB	questions	23K
FTB	newspaper	573K
GSD	news, reviews and Wikipedia	400K
ParTUT	talks, legal texts and Wikipedia	28K
PUD	news and Wikipedia	24K
Sequoia	medical, news and Wikipedia	70K
Spoken	spoken language	34K
Total		1152K

Table 4.4: Summary of used French UD treebanks

4.2.2 Wikipedia

To extract homogeneous data from wikipedia we used the random method for wikipedia pages, provided by the python wikipedia library(wikipedia API wrapper for python). We added a disambiguation step to make sure the suggested pages could be accessed. Then we stored page content into text files after a preliminary cleansing. We followed exactly the same steps for English and French wikipedia.

```
def disambiguate(page_name):
    try:
        wikipedia.page(page_name)
        return page_name
    except wikipedia.exceptions.DisambiguationError as e:
        s = random.choice(e.options)
        return s
    except wikipedia.PageError:
        return None

def get_random_pages_summary(pages=0):
    page_names = [wikipedia.random(1) for _ in tqdm(range(pages))]
    for page_name in page_names:
        if (f := disambiguate(page_name))is not None:
            contents = wikipedia.page(disambiguate(page_name)).content
            contents = re.sub('=+\s*.+\s*=+', '', contents)
            with open(str(page_name+".txt"), "w",encoding="utf-8") as file:
                file.write(contents)
```

4.3 METRICS AND MEASURES

In this section, we will briefly explain the metrics explored for the realization part of the project.

Ideally, a parser's output is evaluated against at least one reference gold standard treebank. Unfortunately though, the cost of the development of domain-specific corpora is really high to allow it.

As mentioned in the introduction, for the purposes of this project, we focus on one question. How can we evaluate out-of-domain data that do not have a gold standard parse tree to test against?

The metrics and measures concerning the parser evaluation on out-of-domain unannotated data that are presented in this report are the following:

- **LAS**
  Labeled attachment score is one of the most common methods for evaluating dependency parsers along with UAS. It considers how many words have been assigned both the correct syntactic head and the correct dependency relation.
- **UAS**
  Unlabeled atatchment score, unlike LAS, only considers the correctness of the assigned syntactic head and ignores the label.
- **POS**
  Parts of speech tagging (POS) is the assignment of parts of speech to individual words in a sentence, i.e at the token level. While processing natural language, POS tagging can be really useful since it helps identify the part of speech of a specific word or token whose POS tag can vary depending on the context.
- **IAA**
  In the introduction we mentioned that we can assume that if two or more parsers agree on the assigned dependencies, then the annotation is reliable. The simplest measure to examine the agreement between two annotators is the Inter-annotator agreement. IAA measures the percentage of observed agreement between two or more annotators. The percentage represents the division of the number of identical annotations by the total number of annotations.

This measure allows us to presume two things. If the annotators agree in most of the cases, we can assume that the annotation guidelines were clear. Secondly, it allows us to make assumptions on how trustworthy the annotation is.

4.4 INTER-PARSER AGREEMENT EVALUATION

We tried to investigate what does consensus between parsers, considering the fact that they have been trained on different training data, reveals about the parsing accuracy.

We report quantitative results as well as a qualitative analysis of the experiment results to tease apart agreements in case of parsing that matches the gold parse and vice versa.

We conduct an experiment on inter-annotator agreement for POS tagging and dependency parsing. We measure both parsing performance and inter-annotator agreement, using tagging and parsing evaluation metrics that are previously explained. This choice allows for a direct comparison between parsing and agreement results. We use Pearson Correlation Coefficient (Pearson's **r**) for this purpose.

For each pair of parsers we parsed the entire dataset with both parsers and reported percentage agreements both in macro average and weighted average. We decided to use both metrics because label classes are very different in size, and macro average might be a good help in order to reflect the actual agreement between the two parsers.

We also compared the parsing result with the gold standard annotation which is available for all selected UD datasets. We evaluated each parser against gold data and computed the agreement with every other parser at the sentence level. Afterwards, we calculated Pearson correlation coefficient for the results obtained from the previous steps. Pearson r is a good choice for our experiments because it measures the significance of the directed relation between values for parse agreements and their accuracy.

We report the correlation between the inter parser agreement and parsing accuracy for all the selected UD datasets and all the parsers mentioned in previous sections. In each experiment one parser is considered as the reference parser, meaning that the output parse from this parser is used as gold parse, even though it might not be completely accurate. The second parser is considered as the one to be evaluated.

To investigate in which setting high agreement might entail high parsing accuracy, correlation coefficient was computed for the reported pairwise agreement metrics and the gold parse evaluation for all selected UD dataset. In each experiment we considered one of the main metrics (POS tagging accuracy, LAS, and UAS) and looked into the correlation between the gold parse evaluation of one parser against its agreement with each one of the other parsers at the sentence level for all the datasets (the reference parser method).

The results for this parts are depicted in scatter plots to make it easier to grasp and compare. We leave the idea of creating different subsets of parsers instead of considering pairwise agreements to future work.

The next part of our experiments was to explore our wikipedia corpora and try to imitate the steps for them with the exception of gold parse evaluation.

We needed a reference parser for this part of our experiments, and we chose it according to the parsing accuracy obtained in the first part of the experiments. We chose our most accurate parsers and used them to parse our un-annotated corpora. Subsequently, we set this parse as gold parse and compare it to another parse given from our second best parser. Following our previous experiments, we tried to inspect the agreement between the two parsers at a sentence level and compared it with the parsing accuracy regarding the gold parse produced by our best parser.

Chapter 5

Results

In this chapter we will present the results observed from the experiments conducted.

In the following sections, we report the correlation matrices of the three metrics, POS, UAS and LAS on the English and French datasets, as well as the evaluations on gold data and scatter graphs illustrating the agreement between parsers compared to the agreement with gold-parses.

5.1 Results obtained from English datasets

Looking into the datasets our parsers have been trained on (presented in 2.2.1) allows us to conclude how different or similar the parsers are. Since we are aware of these characteristics, we can derive some useful information from the tables and figures below.

Probably the most interesting piece of information these experiments provide, is that they reveal that the more different the parsers are, the more correlated their agreement is with the gold evaluation. If we take a step back and look at the data, this seems like a logical conclusion, since it is normal behavior for two parsers who have been trained on similar data, to make similar mistakes. We should expect the parsers to make equally wrong choices. In this case, parsers have a high agreement but accuracy is actually not very high. When we use two very different parsers and we notice an agreement, it is very likely that both are right because they made the same decision based on two different training settings, while when they do not agree, we could consider the possibilities that either one of them is wrong or that both of them are wrong but they have produced the incorrect parse from two different perspectives.

This point will be made clearer with an example. Let us look into two different parsers in particular, LinES and GUM and see how they behave individually but also when used to assess the performance of each other. At figure 3.1 we evaluate their performance on gold data and observe that both parsers' performance is really high. We also know that they are different in terms of training, since their training data are of different size and domain. In this case, we have two equally strong, different parsers. Returning to tables 3.1, 3.2 and 3.3, we can notice their cross-correlation and immediately conclude that we have similar results when using one to assess the performance of the other respectively.

	combined	ewt	gum	linES	partut
combined		0.268 0.322	0.218 0.263	0.392 0.380	0.279 0.256
ewt	0.258 0.315		0.263 0.311	0.401 0.376	0.303 0.278
gum	0.300 0.503	0.310 0.505		0.553 0.566	0.525 0.494
linES	0.401 0.598	0.392 0.587	0.473 0.568		0.493 0.514
partut	0.503 0.662	0.501 0.658	0.475 0.570	0.519 0.566	

Table 5.1: Correlation matrix of "POS" metric on the English datasets. Top value is macro average, bottom is weighted average.

	combined	ewt	gum	linES	partut
combined		0.271 0.274	0.340 0.305	0.302 0.323	0.283 0.284
ewt	0.400 0.411		0.442 0.395	0.374 0.382	0.341 0.335
gum	0.480 0.526	0.465 0.510		0.318 0.398	0.383 0.398
linES	0.494 0.575	0.498 0.559	0.375 0.467		0.465 0.497
partut	0.476 0.555	0.471 0.533	0.452 0.477	0.480 0.497	

Table 5.2: Correlation matrix of "UAS" metric on the English datasets. Top value is macro average, bottom is weighted average.

	combined	ewt	gum	linES	partut
combined		0.356 0.343	0.435 0.386	0.445 0.416	0.365 0.348
ewt	0.430 0.447		0.487 0.450	0.445 0.446	0.403 0.398
gum	0.587 0.608	0.555 0.576		0.495 0.503	0.466 0.465
linES	0.619 0.656	0.588 0.636	0.551 0.570		0.533 0.550
partut	0.608 0.670	0.593 0.652	0.582 0.606	0.562 0.593	

Table 5.3: Correlation matrix of "LAS" metric on the English datasets.
Top value is macro average, bottom is weighted average.

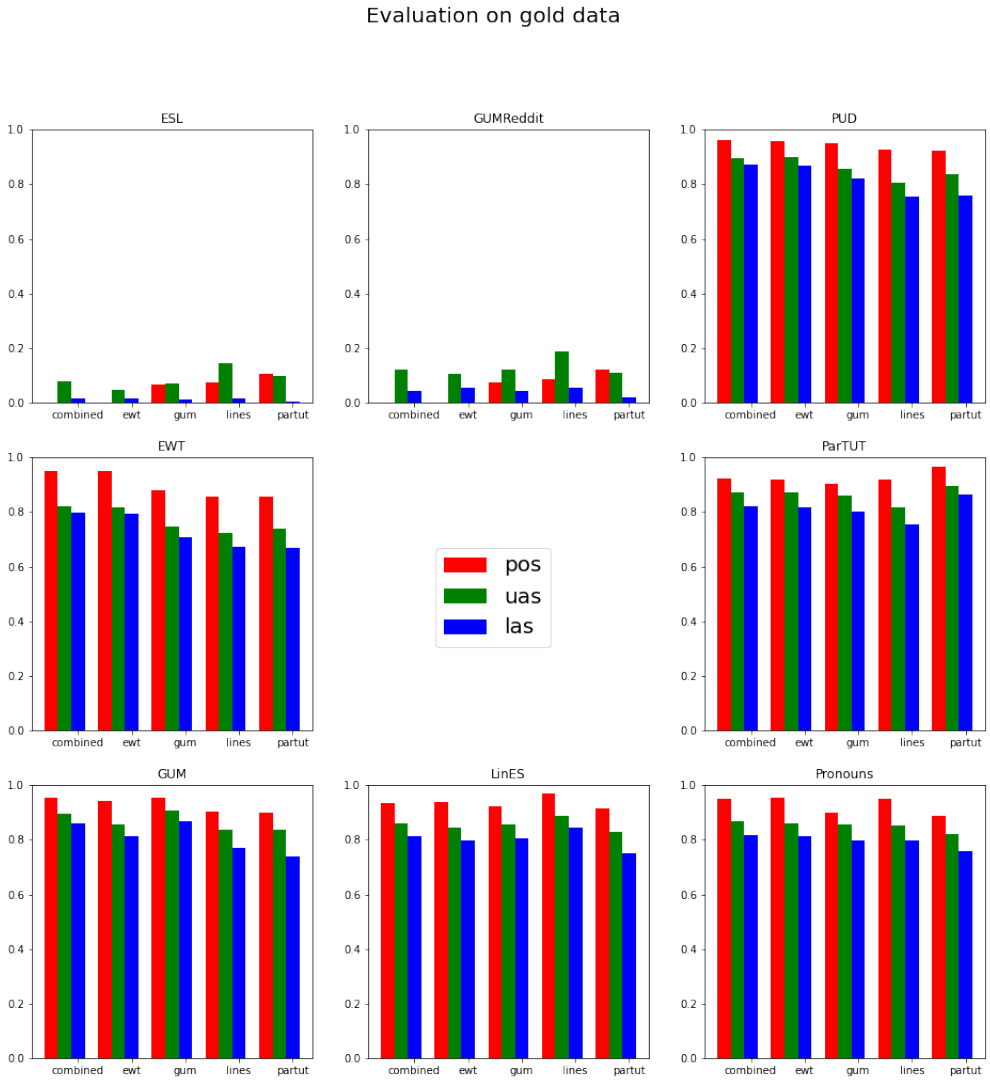
Evaluation on gold data
ESL
GUMReddit
PUD
EWT
ParTUT
GUM
LinES
Pronouns
combined
ewt
gum
lines
partut
pos
uas
las

Figure 5.1

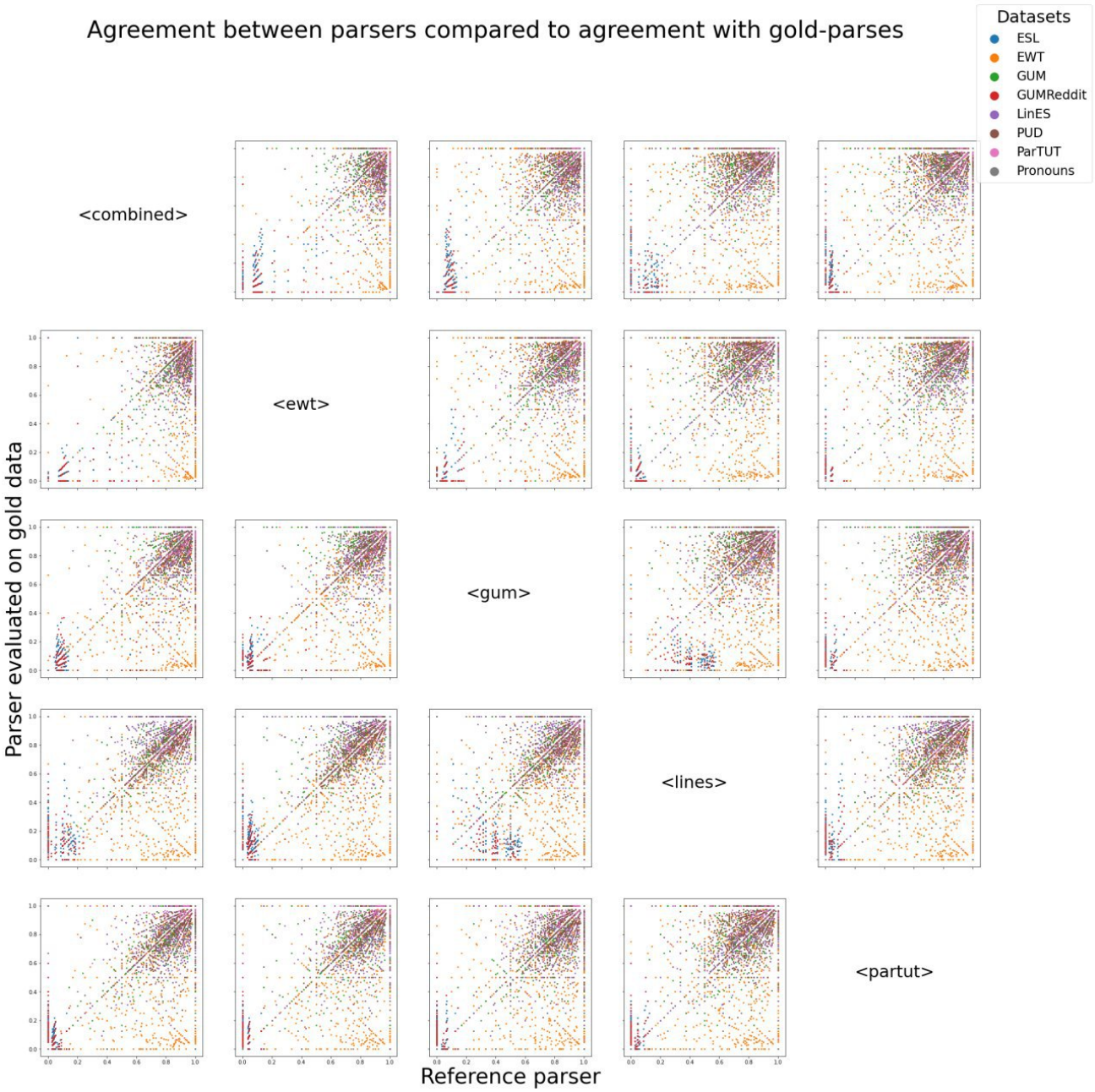
Agreement between parsers compared to agreement with gold-parses
Datasets
ESL
EWT
GUM
GUMReddit
LinES
PUD
ParTUT
Pronouns
<combined>
<ewt>
<gum>
<lines>
<partut>
Parser evaluated on gold data
Reference parser

Figure 5.2

5.2 RESULTS OBTAINED FROM FRENCH DATASETS

The results obtained from the French datasets do not provide any additional useful information but they come to validate the assumption made in the previous section. When observing the tables, we do not notice a strong cross-correlation between a pair of parsers and we could attribute that to the similarity among the four parsers. Their training data are similar and other than GSD, they all share a similar size.

	gsd	partut	sequoia	spoken
gsd		0.372 0.390	0.390 0.281	0.231 0.227
partut	0.728 0.773		0.693 0.715	0.385 0.401
sequoia	0.485 0.371	0.413 0.402		0.313 0.329
spoken	0.782 0.789	0.628 0.559	0.761 0.769	

Table 5.4: Correlation matrix of "POS" metric on the French datasets.
Top value is macro average, bottom is weighted average.

	gsd	partut	sequoia	spoken
gsd		0.295 0.330	0.268 0.254	0.363 0.412
partut	0.535 0.620		0.472 0.565	0.419 0.485
sequoia	0.386 0.416	0.306 0.341		0.394 0.425
spoken	0.533 0.612	0.379 0.425	0.519 0.573	

Table 5.5: Correlation matrix of "UAS" metric on the French datasets.
Top value is macro average, bottom is weighted average.

	gsd	partut	sequoia	spoken
gsd		0.392 0.468	0.366 0.346	0.365 0.440
partut	0.591 0.702		0.528 0.640	0.422 0.532
sequoia	0.453 0.463	0.374 0.433		0.374 0.437
spoken	0.609 0.710	0.480 0.582	0.572 0.659	

Table 5.6: Correlation matrix of "LAS" metric on the French datasets.
Top value is macro average, bottom is weighted average.

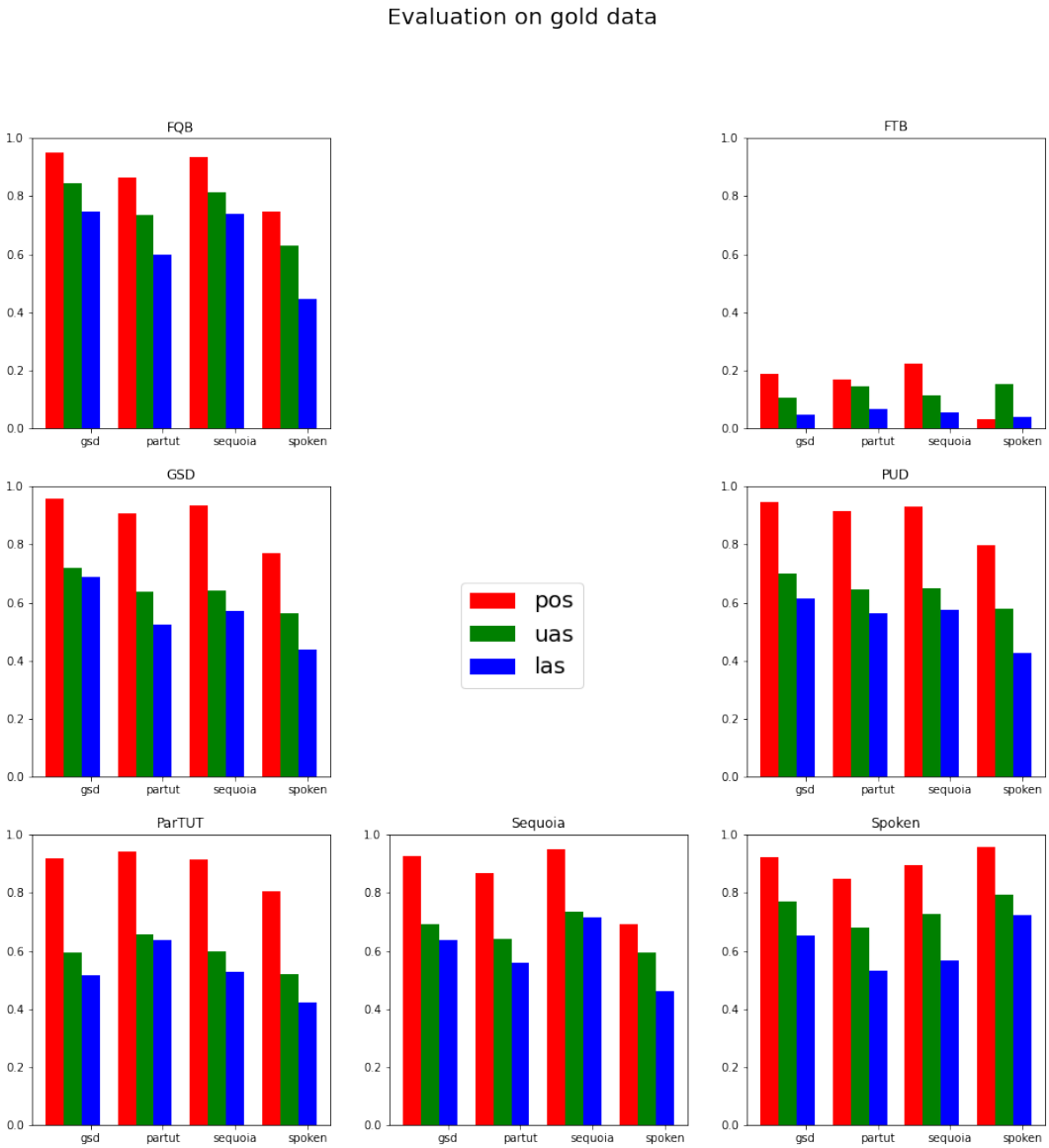
Evaluation on gold data
FQB
FTB
GSD
PUD
ParTUT
Sequoia
Spoken
pos
uas
las
gsd
partut
sequoia
spoken
0.0
0.2
0.4
0.6
0.8
1.0

Figure 5.3


Agreement between parsers compared to agreement with gold-parses
Datasets
FQB
FTB
GSD
PUD
ParTUT
Sequoia
Spoken
<gsd>
<partut>
<sequoia>
<spoken>
Parser evaluated on gold data
Reference parser

Figure 5.4

5.3 RESULTS FOR WIKIPEDIA DATA

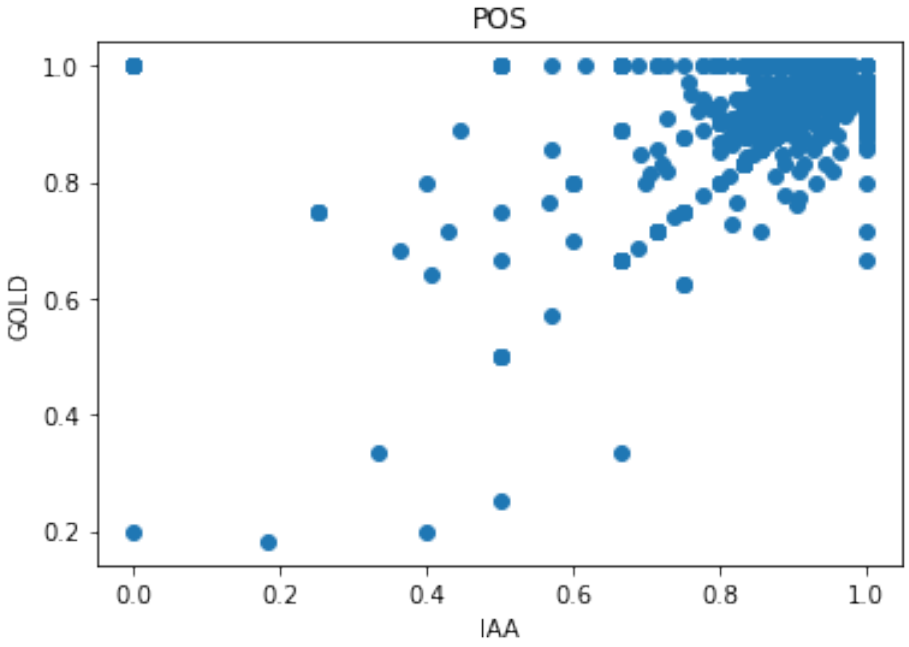

Figure 5.5: English wikipedia data POS agreement compared to gold

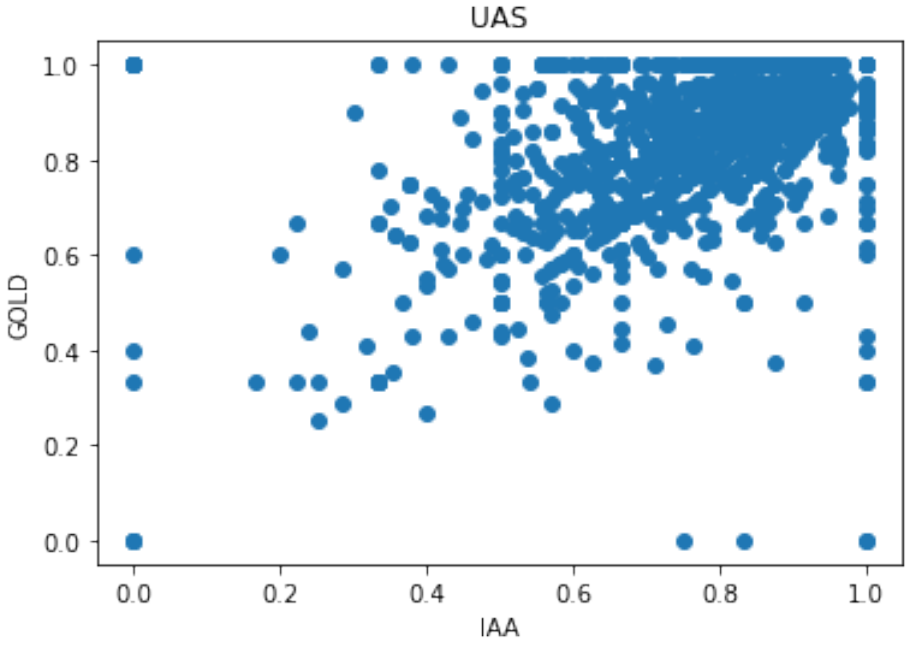

Figure 5.6: English wikipedia data UAS agreement compared to gold

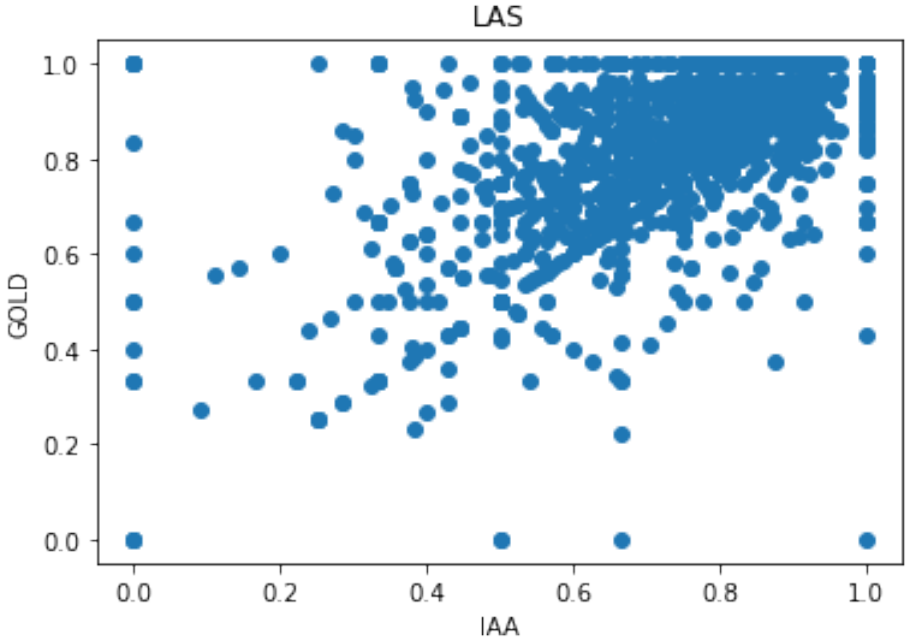

Figure 5.7: English wikipedia data LAS agreement compared to gold

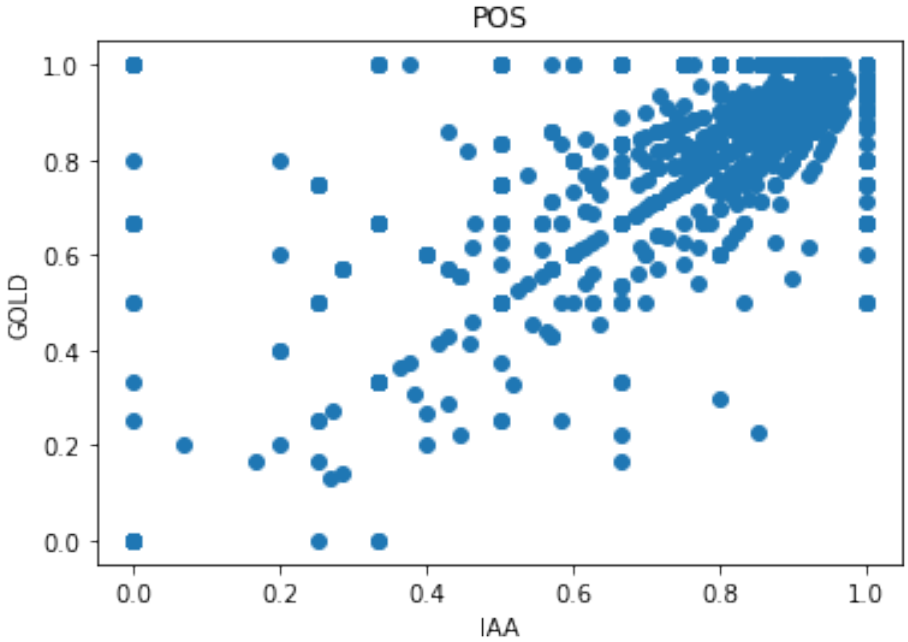

Figure 5.8: French wikipedia data POS agreement compared to gold

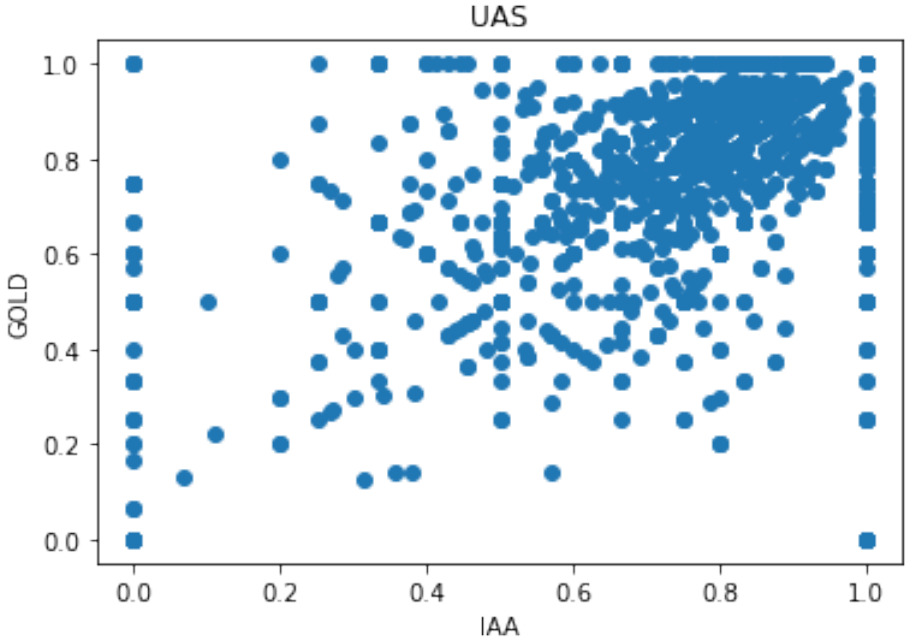

Figure 5.9: French wikipedia data UAS agreement compared to gold

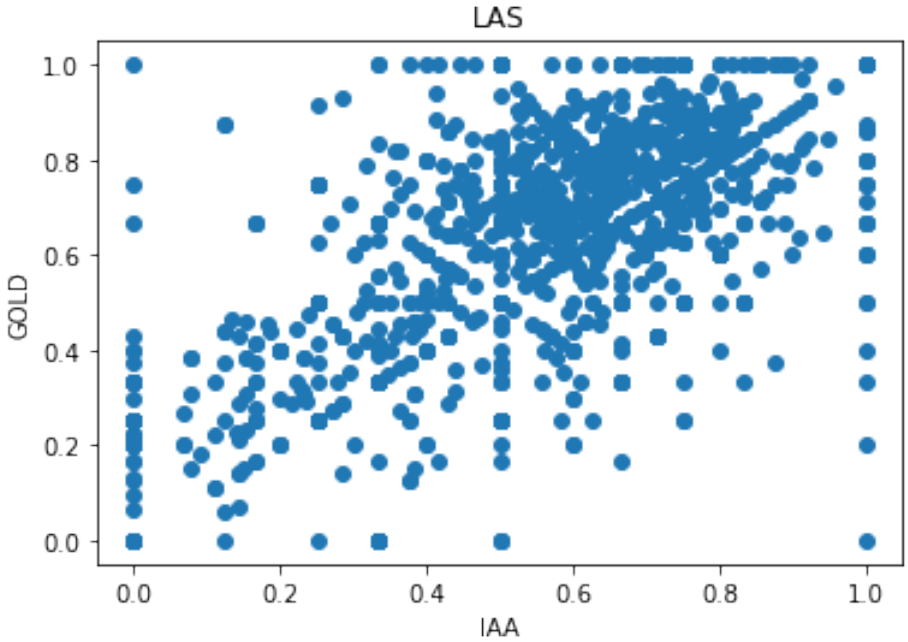

Figure 5.10: French wikipedia data LAS agreement compared to gold

Chapter 6

Conclusion

Experimenting with different datasets and parsers trained on different data allowed us to grasp the concept of this open-research question and the issues that arise from it. Nevertheless, despite our efforts to evaluate inter-parser agreement, more research and experiments are needed to achieve greater results.

In conclusion, we would like to present a few ideas that we think might be of relevance for future work.

In our work, we only compare the agreement of two parsers to the gold evaluation of one of those two. An idea that could prove to be reliable but requires plenty of experiments, would be to consider the accuracy of one parser and try to correlate that to the agreement between three different parsers. In that case, we could use the agreement as a way to assess the performance.

Lastly, we could look deeper into how the different pairs of parsers, for instance those that give the desired results, behave with respect to the different tasks, i.e., POS tagging, and labeled and unlabeled dependency annotation, so that we could make some reliable assumptions. More specifically, if we can prove that a pair of parsers behaves in a stable way among all different tasks, we can estimate how reliable a pair of parsers is and then use it in cases where labeled data are not adequate. For instance, if a corpus provides only POS tagging and not labeled dependencies, we could still use the reliable set of parsers on the corpus since it would have been empirically proven that if a pair performs well on one task, it also performs well on the other ones.

Bibliography

[1] Joakim Nivre et al. “Universal Dependencies v1: A Multilingual Treebank Collection”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 1659–1666. URL: https://www.aclweb.org/anthology/L16-1262.

[2] Sujith Ravi, Kevin Knight, and Radu Soricut. “Automatic prediction of parser accuracy”. In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. 2008, pp. 887–896.

[3] Juri Opitz and Anette Frank. “Automatic accuracy prediction for AMR parsing”. In: *arXiv preprint arXiv:1904.08301* (2019).

[4] Hady Elsahar and Matthias Gallé. “To Annotate or Not? Predicting Performance Drop under Domain Shift”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2163–2173. DOI: 10.18653/v1/D19-1222. URL: https://www.aclweb.org/anthology/D19-1222.

[5] Peng Qi et al. “Stanza: A Python natural language processing toolkit for many human languages”. In: *arXiv preprint arXiv:2003.07082* (2020).

[6] Yevgeni Berzak et al. “Universal Dependencies for Learner English”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 737–746. DOI: 10.18653/v1/P16-1070. URL: https://www.aclweb.org/anthology/P16-1070.

[7] Natalia Silveira et al. “A Gold Standard Dependency Corpus for English”. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May

2014, pp. 2897–2904. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1089_Paper.pdf.

[8] Amir Zeldes. “The GUM Corpus: Creating Multilayer Resources in the Classroom”. In: *Language Resources and Evaluation* 51.3 (2017), pp. 581–612. DOI: http://dx.doi.org/10.1007/s10579-016-9343-x.

[9] Lars Ahrenberg. “LinES: An English-Swedish Parallel Treebank”. In: *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*. Tartu, Estonia: University of Tartu, Estonia, May 2007, pp. 270–273. URL: https://www.aclweb.org/anthology/W07-2441.

[10] Manuela Sanguinetti and Cristina Bosco. “PartTUT: The Turin University Parallel Treebank”. In: *Studies in Computational Intelligence* 589 (Jan. 2015), pp. 51–69. DOI: 10.1007/978-3-319-14206-7_3.

[11] Robert (Munro) Monarch. “Diversity in AI is not your problem, it’s hers”. In: (Aug. 2020). URL: https://medium.com/@robert.munro/bias-in-ai-3ea569f79d6a.

[12] Joakim Nivre et al. “Universal Dependencies 1.2”. In: (2015).

[13] Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. *Application of Graph Rewriting to Natural Language Processing*. Wiley Online Library, 2018.

[14] Djamé Seddah and Marie Candito. “Hard time parsing questions: Building a questionbank for french”. In: *Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. 2016.

[15] Anne Abeill ’e, Lionel Cl é ment, and Lo " i c Li ’e geois. “A corpus arbor ’e for the French c c ais: the French Treebank”. In: *Automatic Language Processing* 60.3 (2019), pp. 19–43.

[16] Bruno Guillaume, Marie-Catherine de Marneffe, and Guy Perrier. “Conversion and improvement of corpus from French annot ’e s to Universal Dependencies”. In: *Automatic Language Processing* 60.2 (2019), pp. 71–95.

[17] Manuela Sanguinetti and Cristina Bosco. “Parttut: The turin university parallel treebank”. In: *Harmonization and development of resources and tools for italian natural language processing within the parli project*. Springer, 2015, pp. 51–69.

[18] Daniel Zeman et al. “CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies”. In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*. 2018, pp. 1–21.

[19] Daniel Zeman et al. “CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies”. In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–19. DOI: 10.18653/v1/K17-3001. URL: https://www.aclweb.org/anthology/K17-3001.

[20] Anne Lacheret et al. “Rhapsodie: a prosodic-syntactic treebank for spoken french”. In: *Language Resources and Evaluation Conference*. 2014.