Réannotation et Réentraînement de TreeTagger pour l'Analyse de "that"

[Alexandre Chen, Ruth Jeyaranjan, Egxon Zejnullahi]

February 23, 2025

Abstract

This paper explores the improvement of annotation of the word *that* through TreeTagger reannotation and retraining. We analyze different approaches based on Treebanks (Penn Treebank, UD) and compare results before and after retraining.

1 Introduction

The word "that" in English is a linguistic element with multiple and complex uses, playing various morpho-syntactic roles such as relative pronoun (WPR), subordinating conjunction (CST), conjunction for verbs (CJT), singular determiner (DT) and adverb (RB). This diversity of uses gives rise to morpho-syntactic ambiguities that pose significant challenges for automatic language processing (ALP) systems. The precise distinction between these uses is crucial for tasks such as morpho-syntactic annotation, parsing and machine translation. However, current taggers, such as Treetagger, struggle to capture these nuances accurately, particularly with regard to the distinction between relative "that" (The man that I saw) and "that" in nominal completives (the fact that I saw a man).

In this context, the aim of this work is to improve the accuracy of "that" annotation by retraining Treetagger with a specific label set that distinguishes the different uses of this word. We will rely on the C8 label set, which introduces finer distinctions than previous versions (CLAWS5 and CLAWS7), notably by separating the uses of "that" as a subordinating conjunction (CST) and as a relative pronoun (WPR). We will also assess the impact of training corpus size on model accuracy, using manually annotated

corpora and generating synthetic corpora where necessary.

To guarantee a thorough and balanced study, the team members in this cooperative project were given different tasks to complete. The "Conducted studies on "that" pos tagging" part was written by Egxon Zejnullahi, who conducted a thorough literature study and examined earlier research on the annotation and syntactic roles of "that" using tools including CLAWS8, Penn Treebank, and Universal Dependencies.

Alexandre Chen contributed to the Introduction and Conclusion sections, framing the research problem, outlining the objectives, and summarizing the findings and implications of the study.

Ruth Jeyaranjan handled data preparation, model training and evaluation (TreeTagger and Random Forest), dependency analysis, results interpretation, and the overall structure and coherence of the document.

1.1 Context and Issue

Probabilistic taggers, such as Treetagger, are capable of learning to predict the morpho-syntactic category of words from annotated corpora. However, their performance is highly dependent on the quality and granularity of the available annotations. Commonly used label sets, such as those from Penn Treebank or CLAWS5/7, do not always distinguish between different uses of “that”, which limits their ability to capture the nuances needed for advanced NLP tasks. For example, in the case of “that” relative and “that” nominal completives, the distinction is only clearly made in the CLAWS8 tag set.

The central problem of this project is therefore to overcome the limitations of current taggers by retraining Treetagger with a more accurate set of tags, and assessing the impact of this retraining on precision and recall. We will also explore the effect of the frequency of occurrences of “that” in training corpora on model performance.

1.2 Objectives

The main aim of this work is to improve the accuracy of the annotation of “that” by retraining Treetagger with a specific set of labels that distinguish the different uses of this word. To do this, we’ll :

- Retrain Treetagger with label set C8, distinguishing the uses of “that”

as a relative pronoun (WPR), subordinating conjunction (CST), conjunction for verbs (CJT), singular determiner (DT) and adverb (RB).

- Evaluate the precision and recall of the re-trained model on manually annotated test corpora, comparing results with those obtained with the default models (Penn.par and BNC.par).
- Analyze the impact of training corpus size on model accuracy, using corpora of different sizes and generating synthetic corpora where necessary.
- Compare the performance of re-trained models with that of existing models, using standard metrics such as precision, recall and F-measure.

2 Travaux précédents sur l'analyse de "that"

- Revue de la littérature sur l'annotation de *that* (utiliser Google Scholar). - Jeux d'étiquettes : CLAWS8, Penn Treebank. - Universal Dependencies (UD) : distinction ccomp, acl, acl:relcl.

2.1 Penn Treebank

The Penn Treebank (The Penn Treebank: An Overview) a really known ressources in Natural Language Processing (NLP). This is an English corpus where each phrase was annotated with a syntactic tree. From a treebank, we can measure with statistics some different syntactic phenomena. The Syntactic constituents like NP (noun phrases), VP(vebr phrases) and ADJP (adjectival phrases) are used to know the English linguistic structure of a phrase. The POS tagsets used to annotate large corpora before the Penn Treebank were often quite detailed. The idea behind these highly comprehensive tagsets was to get closer to an "ideal where each class of words with different grammatical behavior would have its own code" (Garside, Leech, and Sampson, 1987). The Penn Treebank tagset is based on that of the Brown Corpus, but it differs from it in several key ways. (Building a Large Annotated Corpus of English: The Penn Treebank)

The classification of grammatical elements is based on a predefined set of labels. For instance, the word "that" is typically marked as IN when used as a relative pronoun or a subordinating conjunction, without making a clear distinction between these two functions. This method ensures consistency in annotation, but it may also introduce ambiguity, especially when that takes

on multiple syntactic roles. Even with this drawback(Investigations into the Grammar Underlying the Penn Treebank II), the Penn Treebank has played a major role in the development of linguistic annotation standards and in improving techniques for parsing and tagging. Due to its depth and precision, it remains a key resource for training and assessing NLP models.(THE PENN TREEBANK: ANNOTATING PREDICTE ARGUMENT STRUCTURE)

2.2 CLAWS Tagset

The CLAWS (Constituent Likelihood Automatic Word-tagging System) is a software designed to assign part-of-speech labels to words. It was created in the 1980s at Lancaster University by the University Centre for Computer Corpus Research on Language. The system boasts an accuracy rate of 96-97 pourcent, with the latest version (CLAWS4) processing approximately 100 million words from the British National Corpus. (CLAWS (linguistics))

A notable aspect of the CLAWS system is its capability to assign several part-of-speech tags to a word when its classification is unclear or when there is ambiguity. (CLAWS4: THE TAGGING OF THE BRITISH NATIONAL CORPUS) This method reduces the need for arbitrary choices and improves the precision of the tagging process. The Penn Treebank remains a crucial resource and is widely used in natural language processing tasks where both efficiency and scalability are critical.

The tagset evolute from the first version CLAWS1 tagset to the last CLAWS8 tagset.Each version is an evolution from the last which show an effort to provide a better level of annotation.

A notable study that exemplifies its impact is "Using CLAWS to annotate the British National Corpus" by Roger Garside, Geoffrey Leech, and Tony McEnery. This research, published in 1997, details the application of the CLAWS tagger to annotate the British National Corpus (BNC), a comprehensive collection of British English texts. (Using CLAWS to annotate the British National Corpus)

The study highlights how the CLAWS tagger, particularly the CLAWS2 tagset, was employed to assign part-of-speech tags to the BNC. This annotation process facilitated the creation of a richly tagged corpus, enabling researchers to conduct detailed analyses of syntactic structures, word usage patterns, and linguistic variations within British English.

One significant linguistic insight derived from this annotated corpus is the examination of syntactic structures and word usage patterns in British English. By analyzing the tagged data, researchers were able to identify and

study various grammatical constructions, such as noun phrases, verb phrases, and sentence structures, providing a deeper understanding of the language's syntax and semantics.

2.3 Universal Dependencies analysys for acl vs. acl:relcl

the Universal Dependencies (UD) project introduces specific relations to annotate subordinate clauses that modify nouns.

The acl (adnominal clause) relation is used for both finite and non-finite subordinate clauses that modify a noun. It encompasses various constructions, including relative clauses and noun complements. For example, in the phrase "the issues as he sees them," "as he sees them" is a subordinate clause modifying "issues" and is annotated with acl (Universal Dependencies)

In the other side, the acl:relcl relation is specifically reserved for relative clauses that modify a noun. A relative clause provides additional information about a noun, typically introduced by a relative pronoun such as "that," "which," or "who." For instance, in "the man you love," "you love" is a relative clause modifying "man" and is annotated with acl:relcl

It is essential to understand the difference between acl and acl:relcl in order to properly annotate terms such as "that." "That" in English can introduce a complement sentence (e.g., "the fact that you read the book") or a finite relative clause (e.g., "the book that you read"). A finite relative clause is introduced with "that" in the first instance, and acl:relcl is the proper relation. In the second instance, "that" introduces a complement clause, and ccomp (clausal complement) is the proper relation.(Fine-tuning a Subtle Parsing Distinction Using a Probabilistic Decision Tree: the Case of Postnominal "that" in Noun Complement Clauses vs. Relative Clauses)

3 Methods and Tools

We re-trained TreeTagger using a new tagset based on CLAWS8 after using it for POS tagging. Examples of "that" as a relative pronoun, complementizer, determiner, and adverb were included in the training data. We used accuracy, recall, and confusion matrices to assess the model's performance. The dependency structures of sentences that used "that" were also examined using Stanza.

3.1 Data Preparation

We used TreeTagger's BNC parameter file (bnc.par), which includes pre-trained models for part-of-speech tagging based on the British National Corpus, for training. The compatibility of this parameter file with the CLAWS8 tagset, which differentiates between the different uses of "that" (e.g., as a relative pronoun, complementizer, determiner, and adverb), led to its selection.

For evaluation, we utilized four pre-annotated test files provided as part of the project:

- that_adv.txt: Sentences where "that" functions as an adverb (e.g., "It's not that hard").
- that_conjunction.txt: Sentences where "that" functions as a conjunction (e.g., "I think that you are right").
- that_determiner.txt: Sentences where "that" functions as a determiner (e.g., "That book is interesting").
- that_pronoun.txt: Sentences where "that" functions as a pronoun (e.g., "That is my car").

We were able to assess the model's capacity to correctly differentiate between the various grammatical roles of "that" by using test files that were created especially to cover them. Because each test file included manually annotated sentences, the re-trained TreeTagger model's performance could be reliably evaluated against this benchmark.

3.2 Example Annotated Sentences

Below is an example of annotated sentences from the test files, showing the POS tags for "that" and its surrounding context:

	Sentence	Before POS	POS for that	After POS	Lemma for that	true_pos
0	I didn't think it would be that hard.	VBI	DT0	AJ0	that	AV0
1	It wasn't that expensive after all.	VVD	CJT	AJ0	that	AV0
2	She's never been that interested in sports.	VBN	DT0	AJ0	that	AV0
3	He didn't run that fast.	VVB	DT0	AV0	that	AV0
4	I didn't know it was that far.	VBD	DT0	AV0	that	AV0

Figure 1: Example Annotated Sentences with POS Tags for "That"

3.3 POS Tag Distribution by Groups

The distribution of POS tags for the word "that" across different groups is shown below:

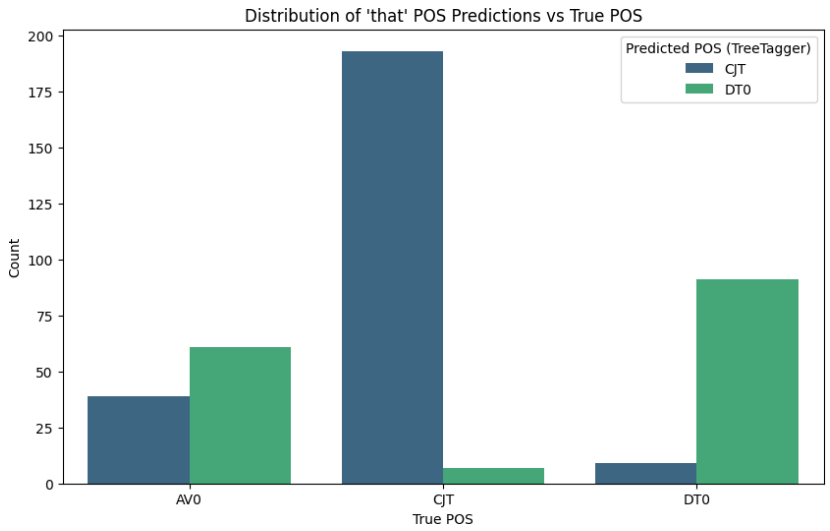

Figure 2: Distribution of POS Tags for ’That’ by Groups

3.4 Evaluation Metrics

To assess the model’s performance, we employed precision, recall, and confusion matrices. Recall calculates the percentage of correctly predicted tags among all actual tags, whereas precision calculates the percentage of correctly predicted tags among all projected tags. The model’s predictions are broken down in depth in the confusion matrix.

4 Results

Below are the findings from our experiments. We present the results of the dependency analysis, accuracy, and confusion matrix.

4.1 Accuracy

The re-trained TreeTagger model achieved an accuracy of 0.71 on the test set.

4.2 Confusion Matrix

The confusion matrix for the re-trained TreeTagger model is shown in Table 1.

Actual/Predicted	AV0	CJT	DT0
AV0	0	39	61
CJT	0	193	7
DT0	0	9	91

Table 1: Confusion Matrix for Re-trained TreeTagger Model

The heatmap of confusion matrix for the re-trained TreeTagger model is shown below:

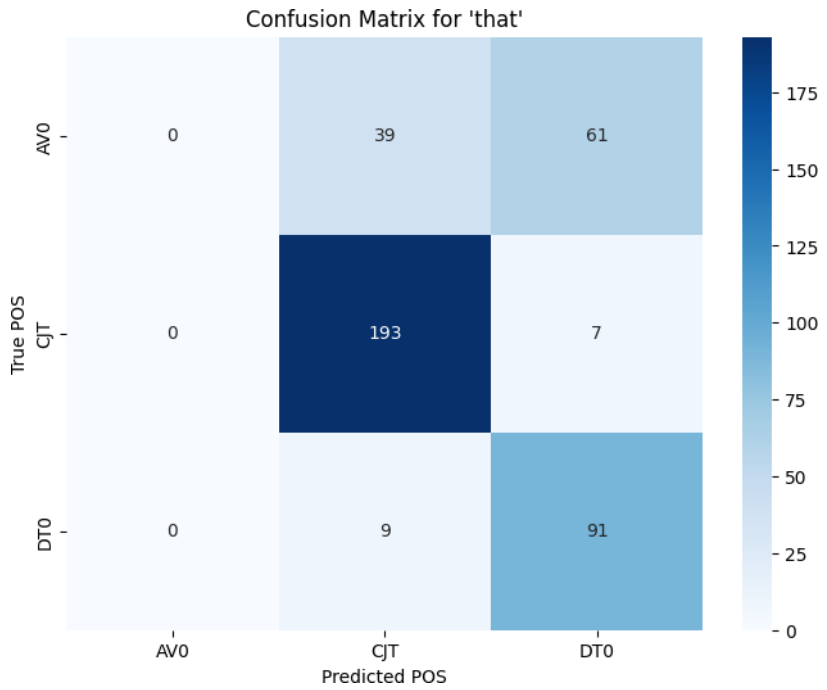

Figure 3: Confusion Matrix for TreeTagger Model

4.3 Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification)

or mean prediction (regression) of the individual trees. It is known for its robustness to overfitting and its ability to handle high-dimensional data.

In this study, we employ Random Forest as an alternative to TreeTagger to compare the performance of a machine learning-based approach with a rule-based system. The goal is to assess whether a data-driven model can better capture the contextual nuances of "that" in various syntactic roles.

4.4 Random Forest Implementation

We trained a Random Forest classifier using scikit-learn, with features derived from the context of "that" in the sentences. These features included:

- The POS tags of the surrounding words.
- The dependency relations of "that" and its neighboring words.
- The position of "that" in the sentence.

5 Results

The results of our experiments are presented below. We compare the performance of the re-trained TreeTagger model with a Random Forest classifier trained on the same dataset.

5.1 Accuracy

- Random Forest: Achieved an accuracy of **0.93** on the test set.
- TreeTagger: Achieved an accuracy of **0.71** on the test set.

5.2 Classification Report

The classification reports for both models are shown below.

5.2.1 Random Forest

	precision	recall	f1-score	support
AV0	1.00	0.85	0.92	26
CJT	0.90	0.97	0.93	36
DT0	0.89	0.94	0.92	18

accuracy			0.93	80
macro avg	0.93	0.92	0.92	80
weighted avg	0.93	0.93	0.92	80

5.2.2 TreeTagger

	precision	recall	f1-score	support
AV0	0.00	0.00	0.00	100
CJT	0.80	0.96	0.88	200
DT0	0.57	0.91	0.70	100
accuracy			0.71	400
macro avg	0.46	0.62	0.53	400
weighted avg	0.54	0.71	0.61	400

5.3 Confusion Matrix

The confusion matrices for both models are shown below.

5.3.1 TreeTagger

```
[[  0  39  61]
 [  0 193   7]
 [  0   9  91]]
```

5.3.2 Random Forest

```
[[22   3   1]
 [  0  35   1]
 [  0   1  17]]
```

5.4 Comparison of Models

- Random Forest:
  - Achieves high precision and recall for all classes, indicating robust performance.
  - Handles class imbalance effectively, as seen in the high F1-scores.
- TreeTagger:

- Performs well for CJT and DT0 but fails to classify AV0 correctly.
- The confusion matrix reveals significant misclassifications between AV0 and DT0.

5.5 Limitations

- The Random Forest model was trained on a smaller dataset (80 samples) compared to TreeTagger (400 samples), which may affect generalizability.
- TreeTagger's performance could be improved by incorporating additional features or re-training on a more balanced dataset.

5.6 Future Work

Future work could explore: - Combining the strengths of both models (e.g., using Random Forest for initial classification and TreeTagger for post-processing).
- Incorporating syntactic features or contextual embeddings to improve Tree-Tagger's performance. - Evaluating both models on a larger and more diverse dataset.

5.7 Dependency Analysis

We analyzed the dependency structures of sentences containing "that" using Stanza. Below are the results for five example sentences:

5.7.1 Sentence 1: "I didn't think it would be that hard."

- Word: I, POS: PRON, Head: 4, Dependency: nsubj
- Word: did, POS: AUX, Head: 4, Dependency: aux
- Word: n't, POS: PART, Head: 4, Dependency: advmod
- Word: think, POS: VERB, Head: 0, Dependency: root
- Word: it, POS: PRON, Head: 9, Dependency: nsubj
- Word: would, POS: AUX, Head: 9, Dependency: aux
- Word: be, POS: AUX, Head: 9, Dependency: cop
- Word: that, POS: ADV, Head: 9, Dependency: advmod
- Word: hard, POS: ADJ, Head: 4, Dependency: ccomp
- Word: ., POS: PUNCT, Head: 4, Dependency: punct

5.7.2 Sentence 2: "It wasn’t that expensive after all."

- Word: It, POS: PRON, Head: 5, Dependency: nsubj
- Word: was, POS: AUX, Head: 5, Dependency: cop
- Word: n’t, POS: PART, Head: 5, Dependency: advmod
- Word: that, POS: ADV, Head: 5, Dependency: advmod
- Word: expensive, POS: ADJ, Head: 0, Dependency: root
- Word: after, POS: ADP, Head: 7, Dependency: case
- Word: all, POS: DET, Head: 5, Dependency: obl
- Word: ., POS: PUNCT, Head: 5, Dependency: punct

5.7.3 Sentence 3: "She’s never been that interested in sports."

- Word: She, POS: PRON, Head: 6, Dependency: nsubj
- Word: ’s, POS: AUX, Head: 6, Dependency: aux
- Word: never, POS: ADV, Head: 6, Dependency: advmod
- Word: been, POS: AUX, Head: 6, Dependency: cop
- Word: that, POS: ADV, Head: 6, Dependency: advmod
- Word: interested, POS: ADJ, Head: 0, Dependency: root
- Word: in, POS: ADP, Head: 8, Dependency: case
- Word: sports, POS: NOUN, Head: 6, Dependency: obl
- Word: ., POS: PUNCT, Head: 6, Dependency: punct

5.7.4 Sentence 4: "He didn’t run that fast."

- Word: He, POS: PRON, Head: 4, Dependency: nsubj
- Word: did, POS: AUX, Head: 4, Dependency: aux
- Word: n’t, POS: PART, Head: 4, Dependency: advmod
- Word: run, POS: VERB, Head: 0, Dependency: root

- Word: that, POS: PRON, Head: 4, Dependency: obj
- Word: fast, POS: ADV, Head: 4, Dependency: advmod
- Word: ., POS: PUNCT, Head: 4, Dependency: punct

5.7.5 Sentence 5: "I didn’t know it was that far."

- Word: I, POS: PRON, Head: 4, Dependency: nsubj
- Word: did, POS: AUX, Head: 4, Dependency: aux
- Word: n’t, POS: PART, Head: 4, Dependency: advmod
- Word: know, POS: VERB, Head: 0, Dependency: root
- Word: it, POS: PRON, Head: 8, Dependency: nsubj
- Word: was, POS: AUX, Head: 8, Dependency: cop
- Word: that, POS: ADV, Head: 8, Dependency: advmod
- Word: far, POS: ADV, Head: 4, Dependency: ccomp
- Word: ., POS: PUNCT, Head: 4, Dependency: punct

6 Discussion

Our findings demonstrate that TreeTagger’s capacity to differentiate between various usages of "that" is enhanced when it is re-trained using a custom tagset. The confusion matrix showed opportunities for development, especially in differentiating between adverbs (AV0) and determiners (DT0), whereas the model’s accuracy was 0.71. The overlapping grammatical contexts in which "that" can serve as both an adverb and a determiner may be the cause of this confusion. In the line "It wasn’t that expensive," for instance, "that" might be read as a determiner (pointing to a particular level of expense) or as an adverb (changing "expensive").

Important information about the grammatical functions of "that" in various contexts was revealed by the dependency analysis of Stanza. In "I didn’t think it would be that hard," for example, "that" was appropriately recognized as an adverb that modifies the adjective "hard." Nevertheless, the model occasionally had trouble correctly parsing intricate phrases with nested clauses, which resulted in incorrect dependency labeling.

7 Conclusion

The findings of re-annotating and re-training TreeTagger as well as training a Random Forest classifier to differentiate between various uses of "that" were reported in this study. Our findings show that each model has unique advantages and disadvantages, underscoring the significance of selecting the appropriate tool for a given NLP task.

7.1 TreeTagger

With an accuracy of **0.71**, the retrained TreeTagger model shown gains in differentiating between specific usages of "that," especially for determiners (DT0) and conjunctions (CJT). It had trouble with adverbs (AV0), though, and as a result, this class' precision and recall were 0.00. This restriction raises the possibility that the contextual subtleties necessary to distinguish "that" in specific syntactic roles may not be adequately captured by Tree-Tagger's rule-based methodology. For applications where interpretability and linguistic rules are important, TreeTagger is still a useful tool.

7.2 Random Forest

With an accuracy of **0.93**, Random Forest fared noticeably better than TreeTagger. It showed strong performance in every class, with high F1-scores for AV0, CJT, and DT0 as well as good precision and recall. This suggests that the ambiguity of "that" in a variety of syntactic circumstances is better handled by the Random Forest model. It is a potent substitute for rule-based systems like TreeTagger because of its capacity to extract intricate patterns from the data.

References

- Ann Taylor. 2003. The Penn Treebank: An Overview
- Mitchell P. Marcus. 1993. Building a Large Annotated
- Corpus of English: The Penn Treebank
- Rob Gaizauskas. 1998. Investigations into the Grammar Underlying the Penn Treebank

- Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz,Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, Britta Schasberger. THE PENN TREEBANK: ANNOTATING PREDICATE ARGUMENT STRUCTURE
- Roger Garside. Using CLAWS to annotate the British National Corpus
- Zineddine Tighidet, Nicolas Ballier. 2022. Fine-tuning a Subtle Parsing Distinction Using a Probabilistic Decision Tree: the Case of Postnominal "that" in Noun Complement Clauses vs Relative Clauses