Exact Grammaticality Judgement of Likely Parse Tree

Jean-Philippe Prost

Exact Grammaticality Judgement of Likely Parse Tree

Jean-Philippe Prost
Jean-Philippe.Prost@lirmm.fr
Research report, 2014

The robustness of probabilistic parsing generally comes at the expense of grammaticality judgement – the grammaticality of the most probable output parse remaining unknown. Parsers, such as the Stanford or the Reranking ones, can not discriminate between grammatical and ungrammatical probable parses, whether their surface realisations are themselves grammatical or not. In this paper we show that a Model-Theoretic representation of Syntax alleviates the grammaticality judgement on a parse tree. In order to demonstrate the practicality and usefulness of an alliance between stochastic parsing and knowledge-based representation, we introduce an exact method for putting a binary grammatical judgement on a probable phrase structure. We experiment with parse trees generated by a probabilistic parser. We show experimental evidence on parse trees generated by a probabilistic parser to confirm our hypothesis.

1 Introduction

The current state-of-the-art parsers[1] generally rely on methods of probabilistic approximation, which make them very robust — in the usual sense in parsing, which refers to the ability of generating a parse for any input sentence, regardless of whether this sentence is well-formed or not. However, that ability comes along with the inability to judge the binary grammaticality of the input sentence. The reason for that is that an optimal parse, even though grammatical, is never assigned maximum probability, which prevents any exact binary judgement based on likelihood. Yet that lack of judgement may be an impediment to various applications. A body of works stress that impediment, for instance, in Natural Language Generation (Wan et al., 2005), Statistical Machine Translation (Zwarts and Dras, 2008), or proof-reading (Bender et al., 2004; Tetreault and Chodorow, 2008). Various techniques are, thus, put together, which aim to combine robust parsing with grammaticality judgement. Some are knowledge-based, while others rely on probabilistic approximation.

First, we quickly review the related literature, and introduce the existing methods, whether exact or stochastic, with respect to grammaticality judgement. Second, we give a quick reminder on Model-Theoretic Syntax, and show how it may be combined with a probabilistic parser in order to solve the issue of judging the grammaticality of a likely structure. Third, we detail the model checking process whereby a structure is judged with respect to a model-theoretic grammar. In a fourth section we experiment our work hypothesis with the Stanford Parser and the binary classification of its output. We conclude in the last section, and discuss further works regarding potential improvements of probabilistic parsers.

2 Grammaticality judgement in the literature

Judgement by likelihood approximation A solution commonly investigated relies on machine learning, and the statistical classification of likely parses. Foster et al. (2008; Wagner (2012) in a context of grammar error correction (focused on specific error types), or Wong and Dras (2011) in a more general context, train classifiers on several combinations of probabilistic grammars[2], and test them on various

[1] According to the ACL wiki, http://aclweb.org/aclwiki/index.php?title=Parsing_(State_of_the_art) (as of 7 March 2014). Last updated on 28 October 2013.

[2] The combinations under investigation are generally among three types of grammars: a grammatical one, extracted from a reference corpus (British National Corpus or Penn Treebank); an erroneous one, extracted from an artificially distorted corpus; and a hybrid one, coming from the merge of both the standard corpus and its ill-formed version.

grammatical and erroneous corpora. The result is used for the extraction of learning features for training the grammaticality classifier. The binary judgement occurs around a probability threshold, which is determined experimentally. Ferraro et al. (2012) also achieve relatively good results with the introduction of knowledge-based counts as training features. All those works justify their approach by the fact that they are confronted to the "excessive robustness" (dixit Wong and Dras (2011)) of the parsers in use[3], since nearly all[4] input sentences get parsed by these parsers in their standard setting.

The underlying hypothesis shared by these works is that the grammaticality of a parse correlates with its likelihood of occurrence. Charniak and Johnson (2005) somehow make the same when relying on a MaxEnt classifier in order to re-rank the n-best parses generated by the Charniak's PCFG[5] parser, according to new probabilities.

Knowledge-based judgement Most of the works, which rely on knowledge-based approaches implement ad hoc strategies, in order to handle classes of errors (more or less) commonly observed (Gamon et al., 2008; Tetreault and Chodorow, 2008). The use of error grammars, or failure recovering techniques during the parsing process are techniques among the most frequently encountered (Bender et al., 2004; Foster and Vogel, 2004). These usually present the advantage of being considerably more efficient than their stochastic counterparts, but on a limited set of specific grammar problems. The approach does not generalise very well, and any change of usage context, or target language, often requires the revision of a significant amount of material. So, for instance, we may wonder whether a system, which is efficient on human productions, would remain as efficient if used in the context of statistical natural language generation, such as summarisation. Their integration seems difficult in systems, where grammaticality judgement is required as a processing step, and not only at the end of a process. Mutton et al. (2007), for instance, in a context of summarisation, hypothesise that the *fluency* of a sentence may be measured through its grammaticality.

The constraint-based methods Robust parsing may also be achieved through constraint-based processing. Those are much less explored than the stochastic approaches, mostly on the ground of a hard-to-handle combinatorial explosion of the search space. In order to be robust, such a parsing process is required, in the worst-case scenario, to explore the entire tree domain under consideration. In case of an ill-formed input utterance in particular, this requirement quickly makes the constraint solving process intractable. Yet, it is important to stress that what makes constraint-based approaches intractable is the constraint solving process, rather than the constraints themselves. Constraints as such remain a very powerful representation tool, as we will see in section 3. We will use them connected in network, in a descriptive way— that is, that they will not be involved in any kind of constraint solving problem. In our case, the purpose of the constraint network will be to contribute to checking the grammar for a given parse. The parse being given, there is no need anymore to look for an assignment for the constraints variables — the parse, somehow, *is* the assignment.

Hence, we hypothesise that the combination made up of a parse structure and its grammar constraint network is a suitable representation for judging the parse's grammaticality, no matter how that parse structure was generated. As far as we are concerned in this paper, we experiment with the generation — hence the exploration of the search space — by a probabilistic parser. The grammaticality judgement implements a model checking process, where the parse structure is the model to be checked against the grammar. Furthermore, we also argue that such knowledge representation, inclusive of a constraint network, may alleviate the exact resolution of a variety of computational problems.

3 Model-Theoretic Syntax

The formal frameworks based on Model Theory already proved their better suitability than formal languages to representing natural language irregularities, infelicities and other oddities, for reasons which we quickly overview. We, then, show how it suits our purpose.

[3] Actually the Stanford Parser and the Reranking Parser for the cited works.

[4] "close to epsilon.

[5] Probabilistic Context-Free Grammar.

Model-Theoretic Knowledge Representation As far as grammaticality is concerned, a convenient feature that the frameworks for Model-Theoretic Syntax (MTS) show is that the logical description of the linguistic properties of an observed utterance is disconnected from its well-formedness. The objects of study, in the sense of the model theory, are *models* of *theories* expressed in a formal (meta-)language, where a *theory* is a set of statements specified in a formal language, and a *model* of a theory is a structure, which satisfies all the statements of that theory[6]. Hence, the modelling of natural language through the Model Theory relies on the *semantics* of a formal language in order to describe grammar relationships among elements of the syntax structure of an utterance. Those relationships may either hold or fail.

Natural Syntax[7] and Model Theory When the domain of structures concerns natural syntax, we get the following interpretation:

- a *theory* is a set grammar statements, specified as a conjunction $\Phi = \bigwedge_i \phi_i$, where every atom ϕ_i is a logical formula, which puts elements of the structure in a relationship;
- a *structure* is a linguistic structure (e.g. phrase structure, dependency structure, ...).

A *grammar* thus comes down to a (generic) conjunctive formula parameterised by the structure, and a theory is an instance of the grammar for a given structure. For a phrase structure domain, for instance, every ϕ_i holds on constituents (e.g. *In a Noun Phrase in French, the Determiner is precedes the Noun.*). Under such an interpretation, the linguistic structure plays the role of a semantic object, which makes a *theory* true (in which case the structure is deemed a *model* of the theory), or not. That is, the parse structure is the object, which makes the grammar (as a conjunction of statements) true, or not.

We use the Property Grammar frameworks (PG), introduced by Blache (2001), for which Duchier et al. (2009) formulated a Model-Theoretic semantics. PG essentially defines eight relations, called *properties*, on a domain of phrase structure trees :

- Obligation (for heads),
- Constituency (for categories of constituents, which may belong to the same phrase),
- Uniqueness (for categories, which may only occur once within a phrase),
- Linearity (for Linear Precedence, in the sense of ID/LP[8]),
- Requirement (for required co-occurrences),
- Exclusion (for mutual exclusion of co-occurrence between constituents of the same phrase),
- Dependency, and
- Agreement.

An interpretation of the semantics of six of those relations[9]is provided in Table 3.

Model checking and grammaticality judgement Let us first introduce a few definitions, in order to fix the notation in use in the following.

Let $\mathcal{S}$ be a set of words in the target language, and $\mathcal{E}$ a set of labels, which denote morpho-syntactic categories; a *lexicon* is then a subset $V \subset \mathcal{E} \times \mathcal{S}$ (which implicitly assumes that the terminals are POS-tagged words). Let $\mathcal{P}_\mathcal{E}$ be the set of all the possible properties on $\mathcal{E}$; a PG grammar Φ is specified by a pair (P_G, V_G), with $P_G \subseteq \mathcal{P}_\mathcal{E}$.

Let $\tau : s$ be a (phrase structure) tree decorated with labels in $\mathcal{E}$, and whose surface realisation is the string of words s; let Φ^s be an instantiation of Φ for $\tau : s$; $\tau : s$ is a model for Φ^s iff $\tau : s$ makes Φ^s true.

[6]Inspired from the Wikipedia article on Model Theory, as of March 2014.

[7]Natural Syntax: shortcut for "natural language syntax".

[8]ID/LP : *Immediate Dominance / Linear Precedence.*

[9]We intentionally omit Dependency and Agreement, which requires the introduction of typed feature structures, out of scope in this paper.

Obligation	$A : \triangle B$	at least one daughter of A is of category B
Constituency	$A : S?$	the category of every daughter of A must be in S
Uniqueness	$A : B!$	at most one daughter of A is of category B
Linearity	$A : B \prec C$	a daughter of category B precedes a daughter of category C
Requirement	$A : B \Rightarrow C$	a daughter of category B requires a daughter of category C
Exclusion	$A : B \not\Rightarrow C$	daughters of categories B and C may not co-occur within the same phrase

Table 1: Interpretation of usual property types in PG

We denote by $\tau : s \models \Phi^s$ the satisfaction of Φ^s by $\tau : s$. The instantiation Φ^s is also called the *constraint network* associated with $\tau : s$ for the grammar Φ.

Definition 1. $\tau : s$ is grammatical with respect to the grammar Φ iff $\tau : s \models \Phi^s$

Since $\Phi^s = \bigwedge_i \phi_i^s$, Definition 1 means that every instance of property ϕ_i^s of Φ^s for the sentence s must be satisfied for s to be deemed grammatical with respect to the grammar Φ.

The model checking process involves:

- instantiating the grammar Φ for the parse tree $\tau : s$,
- building up the corresponding constraint network Φ^s, and
- checking the truth of every atom ϕ_i^s.

Processing-wise, the existence of a linguistic structure is required prior to checking it against a grammar — which involves constructing the constraint network. Figure 3 exemplifies a phrase structure deemed ungrammatical through model checking.

```
419:  En_effet, sept projets sur quatorze, soit la moitié, ont un financement qui
       n' est toujours pas assuré et dont le calendrier n' est pas_encore arrêté.
( (SENT (ADV En_effet) (PUNC ,)
         (NP (DET sept)
             (NC projets)
             (PP (P sur)
                 (NP (NC quatorze)))
             (PUNC ,)
             (COORD (CC soit)
                 (NP (DET la) (NC moitié)))) (PUNC ,)
         (VN (V ont))
         (NP (DET un)
             (NC financement)
             (Srel
                 (NP (PROREL qui))
                 (VN (ADV n')
                     (V est)
                     (AdP (ADV toujours) (ADV pas))
                     (VPP assuré))))
         (COORD (CC et)
             (Sint
                 (NP (NC dont) (DET le) (NC calendrier))
                 (VN (ADV n') (V est) (ADV pas_encore) (VPP arrêté)))) (PUNC .)))
```

Figure 1: Example of parse deemed ungrammatical through model checking

Generation of candidate-model structures Since a Model-Theoretic representation is independent from any processing aspects concerning the generation of candidate structures, the generation strategy may be designed separately from model checking. Although an inference process may be designed in order to construct structures on the sole basis of the constraint grammar (Maruyama, 1990; Balfourier et al., 2002; Prost, 2008; Duchier et al., 2010, among others), nothing makes it compulsory. It is, therefore,

possible, for instance, to check the likely structures generated by a probabilistic parser against the MT grammar.

Note, as well, that the type of linguistic structure concerned by a Model-Theoretic representation may take different forms, depending on the formal framework in use. The seminal work of Maruyama (1990), for instance, is concerned with dependency structure, while Optimality Theory (Prince and Smolensky, 1993) is more used for describing phonological structures. As for PG, the framework is essentially used with phrase structures, though a few works rely on it for multimodal annotation, or biological sequences analysis.

4 Model checking and grammaticality

As we saw, grammaticality judgement is achieved through model checking. The process requires two resources: a PG grammar, and a parse tree to serve as a candidate-model. The grammar is then instantiated with the input parse in order to construct the constraint network. The grammar we use for experimenting derives from an implicit CFG corpus grammar. What follows describes the derivation process. We do not detail here the instantiation of the grammar for a given parse, since it is only a matter of unifying the constituent categories specified in properties with the labelled nodes in the parse tree. In the literature, the instantiation is usually part of the *characterisation* process, along with the grammar checking of instances. By extension, the term *characterisation* is also used for denoting the resulting constraint network.

4.1 Rule-base derivation of a GP grammar

Given an implicit CFG corpus grammar, a GP grammar may be derived through the application of rules, specific to the semantics of each property type. Most of the rules which we describe next were introduced by Blache and Rauzy (2012).

Let C be the label of a non-terminal node; we denote by R_C the set of context-free rules of left-hand side C, and we define RHS to map every C to the set $\mathrm{RHS}(R_C)$ of its right-hand side unique lists of labels. We also define label to map a node x to its label.

Rule 0 (**Obligation** – not implemented yet)**.** The semantics of Obligation is meant to capture head phrases. Thus theoretically, the Obligation property $C : \triangle H_C$ is specified by the set H_C of disjunctions $\psi = \bigvee e$ of distinct labels e, such that an e occurs in every rule $\mathrm{RHS}(R_C)$. Its implementation requires to address a problem of maximum set cover, which is proven NP-hard.

Further works will implement a greedy algorithm of approximation. Another option could be to heuristically gather heads (), or possibly to combine the two (heuristics and greedy algorithm).

Rule 1 (**Constituency).** The Constituency property $C : E_C?$ is specified for C by the set E_C of distinct labels e, such that there exists $r \in \mathrm{RHS}(R_C)$ with $e \in r$.

$$E_C \equiv \{e, \exists x \exists r (r \in \mathrm{RHS}(R_C)) \wedge (x \in r) \wedge \mathrm{label}(x) = e\}$$

Figure 4.1 exemplifies Constituency properties derived from the French Sequoia treebank (Candito and Seddah, 2012). Each property meets `PHRASE_LABEL: list_of_constituents`.

```
AdP: [DET, Srel, Ssub, NP, ADV, COORD, PP]
SENT: [NC, NP, ADV, VPpart, VN, VPinf, ADVWH, PP, AdP, I, Srel, Ssub, AP, Sint, NPP, COORD]
```

Figure 2: Sample PG Constituency properties derived from the Sequoia corpus

Rule 2 (**Uniqueness).** The Uniqueness property $C : U_C!$ is specified for C by the set U_C of all the distinct labels, which never co-occur with themselves within the same right-hand side.

$$\begin{aligned} U_C \equiv \{&e, \forall x \forall y \forall r \\ &r \in \mathrm{RHS}(R_C) \wedge (x \in r) \wedge (y \in r) \wedge ((\mathrm{label}(x) = \mathrm{label}(y)) \rightarrow (x = y)) \\ &\wedge \mathrm{label}(x) = e\} \end{aligned}$$

```
COORD: [DET, AdP, CC, Srel, Ssub, AP, Sint, VPpart, VPinf, VN]
VPinf: [VPpart, CLO, VPinf, AdP, VINF, Sint]
PP: [PROREL, NC, P, ADJ, VPpart, VPinf, PRO, AdP, P+D, AP, Sint, P+PRO]
```

Figure 3: Sample PG Uniqueness properties derived from the Sequoia corpus

Figure 4.1 exemplifies Uniqueness properties.

Rule 3 (Linearity). The set of Linearity properties $C : a \prec b$ for C is specified by the set L_C of consistent ordered pairs of labels (a, b), where (a, b) is consistent iff there exists a production rule $r \in R_C$, such that a and b co-occur in the right-hand side of r, and there exists no rule $r' \in R_C$ such that $(b, a) \in r'$. We denote by i_x the index of node x in the rule's right-hand side.

$$\begin{aligned} L_C \equiv \big\{&(a,b), \forall x \forall y \forall r \exists x' \exists y' \neg \exists r' \\ &r \in \mathrm{RHS}(R_C) \wedge (x \in r) \wedge (y \in r) \wedge \mathrm{label}(x) = a \wedge \mathrm{label}(y) = b \wedge (i_x < i_y) \\ &\wedge r' \in \mathrm{RHS}(R_C) \wedge (x' \in r') \wedge (y' \in r') \wedge \mathrm{label}(x') = a \wedge \mathrm{label}(y') = b \\ &\wedge (i_{y'} < i_{x'})\big\} \end{aligned}$$

Figure 4.1 exemplifies Linearity properties.

```
VN: [[V, VPP], [VINF, AdP], [CLS, VPP], [V, PP], [CLS, ADV], [CLO, VS], [CLO, VINF], [PP, ADV],
    [CLO, VPP], [CLR, VPP], [VS, VINF], [VS, VPP], [CLS, VS], [PP, VPP], [V, COORD], [V, AdP],
    [VPR, VINF], [V, NP], [CLR, VS], [CLO, V], [VIMP, CLO], [VPR, VPP], [CLR, CLO], [CLO, VPR],
    [AdP, VPP], [CLO, AdP], [V, VINF], [CLR, VPR], [CLS, AdP], [NP, VPP]]
COORD: [[CC, VN], [CC, PP], [ADV, Ssub], [VPpart, NP], [Sint, PP], [CC, VPpart], [AdP, NP],
    [CC, VPinf], [CC, AP], [CC, Sint], [VN, Ssub], [CC, Srel], [CC, Ssub], [CC, AdP], [VN, AP],
    [CC, ADV], [ADV, PP], [VN, VPinf], [CC, DET], [AP, VPinf], [ADV, VPinf], [AP, PP], [CC, NP],
    [NP, Sint], [VN, VPpart]]
```

Figure 4: Sample PG Linearity properties derived from the Sequoia corpus

Rule 4 (Requirement). First, let us remind that the semantics of the Requirement property $C : x \Rightarrow y$ differs from the semantics of implication, in that unlike implication $(C : x \Rightarrow y) \not\equiv (\neg x \vee y)$. Therefore, the set Z_C of Requirement properties is specified by the set of co-occurrences within the same phrase, minus the co-occurrences for which the left element may also occur without the right element. Thus, among the set of co-occurrences (a, b) for C, we remove those for which there exists a production rule $r \in \mathrm{RHS}(R_C)$ such that $a \in r$ and $b \notin r$.

$$\begin{aligned} Z_C \equiv \{&(a,b), \forall x \forall r \exists y \\ &r \in \mathrm{RHS}(R_C) \wedge (x \in r) \wedge (y \in r) \wedge (x \neq y) \wedge ((\mathrm{label}(x) = a) \rightarrow ((\mathrm{label}(y) = b)))\} \end{aligned}$$

Rule 4 is only an approximation, since it does not capture any disjunctive y token, as allowed by the property's semantics. We discuss consequences of this limitation in section 5, and envisage possible improvements.

Rule 5 (Exclusion). The set of Exclusion properties $C : x \not\Rightarrow y$ is specified by the set X_C of the pairs of labels (a, b), such that a and b never co-occur within the same right-hand side.

$$\begin{aligned} X''_C \equiv \big\{&(a,b), \exists x \exists y \exists r \exists r' \\ &r \in \mathrm{RHS}(R_C) \wedge (x \in r) \wedge \mathrm{label}(x) = a \\ &r' \in \mathrm{RHS}(R_C) \wedge (y \in r') \wedge \mathrm{label}(y) = b\big\} \\ X'_C \equiv \{&(a,b), \exists x \exists y \exists r \\ &r \in \mathrm{RHS}(R_C) \wedge (x \in r) \wedge (y \in r) \wedge (x \neq y) \wedge \mathrm{label}(x) = a \wedge \mathrm{label}(y) = b\} \\ X_C \equiv &X''_C \backslash X'_C \end{aligned}$$

Rule 5 might be excessively restrictive, as it enumerates the exhaustive list of forbidden co-occurrences. Yet, at this stage, we can not think of a better way of handling the derivation of this property.

5 Experiments: grammaticalisation of candidate-parses

We experimented with the Stanford Parser (STP) (Green et al., 2011) and the French Sequoia treebank (Candito and Seddah, 2012, hereafter CS2012), in order to determine the impact of grammaticality judgement by model checking on parsing. In Experiment 1 we judge the grammaticality of the parses generated by the STP. In Experiment 2 we judge the grammaticality of the n-best candidate-parses generated by the STP for every input sentence, then we re-rank the candidates according to their grammaticality (the grammatical ones come first). In both experiments the parser's train set is the same as in CS2012, which is also used for deriving the PG grammar.

PG grammar derivation The PG grammar, which we used in all our experiments derives from the same train set as in CS2012. Table 5 reports different extraction figures for the implicit CFG grammar, as well as the numbers of PG properties derived by type.

POS-tags	Phrasal rules	TOTAL
5817	1409	7226

Constituency	Uniqueness	Linearity	Requirement	Exclusion
165	108	321	99	678

Table 2: Derivation of a PG grammar from Sequoia (implicit CFG rules and derived properties)

Experiment 1 (Grammar Checking)**.** First, we run the STP on the Sequoia's test set (the PARSEVAL measures[10] are reported in Table 5), then we check the grammar for every parse (*characterisation* process) in order to judge its grammaticality. Some of the negative characterisations, hence ungrammatical phrases/sentences, are reported in Table 5.

Precision	Rappel	FMeasure	Exact matches	Valid sentences / Total
68.51	70.46	69.47	205 (19.65%)	1043 / 1043

Table 3: PARSEVAL measures of the STP on Sequoia

Among 1043 parsed sentences, 36 are judged grammatical. They represent 3.45% of the total, and $36/838 \simeq 4.3\%$ of the incomplete matches. This figure ought to be put in perspective with respect to different elements. First is the syntactic ambiguity inherent to natural language, which may explain that a parse may be grammatical even though it does not exactly match the gold standard. Second is the rather incomplete nature of the PG grammar used for the experiment. It is incomplete for several reasons: (a) the derivation of the Obligation property, which is meant to identify the head constituents, is not implemented yet; (b) the derivation of the Requirement property is not implemented in a sufficiently satisfactory manner, since it does not capture yet the genuine semantics; (c) other properties, such as

[10]Obtained with the programme `evalb`: `http://nlp.cs.nyu.edu/evalb/`, version of 2 November 2013.

Sent.		Rewrite Rule	Property	Violated instances
5	VN →	[ADV, V, AdP, VPP, VPP]	Exclusion	P- = [[ADV, AdP]]
331	Sint →	[VPpart, NP, VN, PP, NP]	Linearity	P- = [[PP, VPpart]]
	Sint →	[VN, PP, Sint, VPpart]	Requirement	P- = [[Sint, NP], [VPpart, NP]]
	→		Exclusion	P- = [[VPpart, Sint]]
502	SENT→	[NP, Sint, VN, VPinf, ADV, Srel]	Linearity	P- = [[VPinf, Sint]]
	→		Exclusion	P- = [[VPinf, Srel], [Srel, Sint]]
1022	SENT→	[NP, VN, AP, NP, Srel]	Exclusion	P- = [[Srel, AP]]

Table 4: Sample characterisation: grammar check yielding (un)grammaticality judgements

Dependency and Agreement are not implemented either, for they require the extraction of typed feature structures. We conjecture that further developments of the grammar derivation process might yield better results.

We also observe — though we have not measured the phenomena — that various patterns show up, which inform on the way solution parses are preferred by the parser. The corpus annotation scheme, for instance, seems to influence the derivation of the model-theoretic grammar, in that some labels are over-specified. The discrimination of Nouns, for example, between Common Nouns (NC) and Proper Nouns (NP) is an impediment to the derivation of a Requirement property meant to specify that within a Noun Phrase a Determiner requires a Noun. We hypothesise that the problem could be addressed with the introduction of feature structures in the grammar, and their extraction from the corpus. But even though, the annotation scheme, this time, will probably show under-specified.

Another quite recurrent pattern is seemingly the choice made by the parser to go perhaps "too easily" for a sub-optimal POS-tagging of the input. Given that the state-of-the-art POS-taggers perform with about 97% accuracy, we speculate that a more sophisticated re-ranking strategy (than the naive one we expose next) could grant a high priority to the candidates based on the initial POS-tagging.

Experiment 2 (Grammaticalisation)**.** This experiment aims to "*grammaticalise*" the STP's output. We proceed as follows:

1. (*n-best parsing*) the n-best candidate-parses are generated with the STP for the sentences deemed ungrammatical in Experiment 1;

2. (*characterisation*) every candidate-parse is characterised, and its grammaticality inferred;

3. (*re-ranking*) the candidates are re-ranked, in order to place the grammatical ones in the first positions;

4. (*evaluation*) the new grammaticalised corpus is evaluated.

Table 5 reports the number of parses being grammaticalised in terms of number n of candidates. Surprisingly, the bracketing scores remain unchanged : $P = 68, R = 70, F = 69$ regardless of n.

n	10	20	30	40	50	60	70	80	100	150	300
grammaticalised parses	13	18	19	22	23	24	24	24	25	26	28

Table 5: Results of the re-ranking by grammaticalisation

6 Conclusion

We have investigated with the alliance between probabilistic parsing and Model-Theoretic Syntax. We have shown that a parse tree may easily be combined with a constraint network through the use of a corpus-based model-theoretic grammar. A benefit of such a combination is that it provides a finer-grained description of the linguistic properties of a sentence than its sole parse structure, and it alleviates the address of various language problems. In order to demonstrate that benefit we have experimented with the grammaticality judgement of probable parses through a process of model checking: a parse structure is deemed grammatical if and only if it satisfies all the grammar properties, i.e. iff it is a model of the conjunction of grammar properties. Our experiments have shown that 3.5% of the Stanford Parser's outcome on the Sequoia corpus of French are deemed ungrammatical. Meanwhile, they have also shown that a naive re-ranking of the n-best candidate-parses according to their grammaticality is not sufficient a strategy to improve the parser's performance. We have discussed a few avenues for further works, which touch both the improvement of the derivation of a model-theoretic grammar, and the sophistication of the re-ranking strategy by model-checking.

References

[Balfourier et al.2002] Jean-Marie Balfourier, Philippe Blache, and Tristan Van Rullen. 2002. From Shallow to Deep Parsing Using Constraint Satisfaction. In *Proc. of the 6th Int'l Conference on Computational Linguistics (COLING 2002)*.

[Bender et al.2004] Emily M. Bender, Dan Flickinger, Stephan Oepen, Annemarie Walsh, and Timothy Baldwin. 2004. Arboretum: Using a precision grammar for grammar checking in CALL. In *Proceedings of InSTIL/ICALL2004–NLP and Speech Technologies in Advanced Language Learning Systems–Venice*, volume 17, page 19.

[Blache and Rauzy2012] Philippe Blache and Stéphane Rauzy. 2012. Enrichissement du ftb: un treebank hybride constituants/propriétés.

[Blache2001] Philippe Blache. 2001. *Les Grammaires de Propriétés : des contraintes pour le traitement automatique des langues naturelles*. Hermès Sciences.

[Candito and Seddah2012] M. Candito and D. Seddah. 2012. Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *Actes de TALN'2012*, Grenoble, France.

[Charniak and Johnson2005] Eugene Charniak and Mark Johnson. 2005. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *ACL*.

[Duchier et al.2009] Denys Duchier, Jean-Philippe Prost, and Thi-Bich-Hanh Dao. 2009. A model-theoretic framework for grammaticality judgements. In *Formal Grammar*, pages 17–30.

[Duchier et al.2010] Denys Duchier, Thi-Bich-Hanh Dao, Yannick Parmentier, and Willy Lesaint. 2010. Property grammar parsing seen as a constraint optimization problem. In *FG*, pages 82–96.

[Ferraro et al.2012] Francis Ferraro, Matt Post, and Benjamin Van Durme. 2012. Judging grammaticality with count-induced tree substitution grammars. In *The Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications*.

[Foster and Vogel2004] Jennifer Foster and Carl Vogel. 2004. Parsing Ill-Formed Text Using an Error Grammar. *Artificial Intelligence Review*, 21(3):269–291.

[Foster et al.2008] Jennifer Foster, Joachim Wagner, and Josef van Genabith. 2008. Adapting a wsj-trained parser to grammatically noisy text. In *ACL (Short Papers)*, pages 221–224.

[Gamon et al.2008] Michael Gamon, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for esl error correction. In *IJCNLP*, volume 8, pages 449–456.

[Green et al.2011] Spence Green, Marie-Catherine de Marneffe, John Bauer, and Christopher D. Manning. 2011. Multiword Expression Identification with Tree Substitution Grammars: A Parsing tour de force with French. In *EMNLP 2011*.

[Maruyama1990] Hiroshi Maruyama. 1990. Structural Disambiguation with Constraint Propagation. In *Proceedings 28th Annual Meeting of the ACL*, pages 31–38, Pittburgh, PA.

[Mutton et al.2007] Andrew Mutton, Mark Dras, Stephen Wan, and Robert Dale. 2007. GLEU: Automatic Evaluation of Sentence-Level Fluency. In *ACL*.

[Prince and Smolensky1993] Alan Prince and Paul Smolensky. 1993. Optimality Theory: Constraint Interaction in Generatire Grammar. Technical report, TR-2, Rutgers University Cognitive Science Center, New Brunswick, NJ.

[Prost2008] Jean-Philippe Prost. 2008. *Modelling Syntactic Gradience with Loose Constraint-based Parsing*. Ph.D. thesis, Macquarie University, Sydney, Australia, and Université de Provence, Aix-en-Provence, France (cotutelle).

[Tetreault and Chodorow2008] Joel R Tetreault and Martin Chodorow. 2008. The ups and downs of preposition error detection in esl writing. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 865–872. Association for Computational Linguistics.

[Wagner2012] Joachim Wagner. 2012. *Detecting Grammatical Errors with Treebank-Induced, Probabilistic Parsers*. Ph.D. thesis, Dublin City University, Dublin, Ireland.

[Wan et al.2005] Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2005. Towards statistical paraphrase generation: preliminary evaluations of grammaticality. In *Proceedings of The 3rd International Workshop on Paraphrasing (IWP2005)*, pages 88–95.

[Wong and Dras2011] Sze-Meng Jojo Wong and Mark Dras. 2011. Exploiting Parse Structures for Native Language Identification. In *EMNLP*, pages 1600–1610.

[Zwarts and Dras2008] Simon Zwarts and Mark Dras. 2008. Choosing the right translation: A syntactically informed classification approach. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1153–1160. Association for Computational Linguistics.