HAL
open science

Using FB-LTAG Derivation Trees to Generate Transformation-Based Grammar Exercices

Claire Gardent, Laura Perez-Beltrachini

To cite this version:

Claire Gardent, Laura Perez-Beltrachini. Using FB-LTAG Derivation Trees to Generate Transformation-Based Grammar Exercices. TAG+11: The 11th International Workshop on Tree Adjoining Grammars and Related Formalisms, Sep 2012, Paris, France. pp.117-125. hal-00768231

HAL Id: hal-00768231

https://hal.science/hal-00768231v1

Submitted on 21 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using FB-LTAG Derivation Trees to Generate Transformation-Based Grammar Exercises

Claire Gardent
CNRS/LORIA
Nancy, France
Claire.Gardent@loria.fr

Laura Perez-Beltrachini
Université de Lorraine/LORIA
Nancy, France
Laura.Perez@loria.fr

Abstract

Using a Feature-Based Lexicalised Tree Adjoining Grammar (FB-LTAG), we present an approach for generating pairs of sentences that are related by a syntactic transformation and we apply this approach to create language learning exercises. We argue that the derivation trees of an FB-LTAG provide a good level of representation for capturing syntactic transformations. We relate our approach to previous work on sentence reformulation, question generation and grammar exercise generation. We evaluate precision and linguistic coverage. And we demonstrate the genericity of the proposal by applying it to a range of transformations including the Passive/Active transformation, the pronominalisation of an NP, the assertion / yes-no question relation and the assertion / wh-question transformation.

1 Introduction

Textbooks for language learning generally include grammar exercises. For instance, *Tex's French Grammar* [1] includes at the end of each lecture, a set of grammar exercises which target a specific pedagogical goal such as the one shown in Figure 1 for learning to form questions. The aim of those exercises is to facilitate the acquisition of a specific grammar point by presenting the learner with *exercises made up of short sentences involving a simple syntax and a restricted vocabulary.*

As argued in (Perez-Beltrachini et al., 2012), most existing work on the generation of grammar exercises has concentrated on the automatic creation of exercises whose source sentences are "real life sentences" extracted from an existing corpus. In contrast, we aim at generating textbook style exercices i.e., exercices whose syntax and lexicon are controlled to match the linguistic content already acquired by the learner.

Moreover, in computer aided language learning (CALL), much of the work towards generating exercices has focused on so-called objective test items i.e., test items such as multiple choice questions, fill in the blank and cloze exercise items, whose answer is strongly constrained and can therefore be predicted and checked with high accuracy. Thus, (Chen et al., 2006) describes a system called FAST which supports the semi-automatic generation of Multiple-Choice and Error Detection exercises while (Aldabe et al., 2006) presents the ArikiTurri automatic question generator for constructing Fill-in-the-Blank, Word Formation, Multiple Choice and Error Detection exercises.

Few studies, however, have been conducted on the generation of transformation based exercices such as illustrated in Figure 1.

In this paper, we present an approach for generating transformation exercices such as (1), where the query (Q) is a sentence and the solution (S) is related to the query by a syntactic transformation.

(1) Instruction: *Modify Q so that the underlined verb is in passive.*
Q: John hopes that Mary <u>likes</u> Peter.

[1] *Tex's French Grammar* http://www.laits.utexas.edu/tex/ is an online pedagogical reference grammar which is arranged like many other traditional reference grammars with the parts of speech (nouns, verbs, etc.) used to categorize specific grammar items (gender of nouns, irregular verbs). Individual grammar items are explained in English, exemplified in a dialogue, and finally tested in self-correcting, fill-in-the-blank exercises.

Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11), pages 117–125, Paris, September 2012.

```
Interrogative pronouns. Transform each sentence into the corresponding question.
Example: Rita parle DE LA POLITIQUE. You write: De quoi est-ce que Rita parle?
1. Bette parle DE TAMMY.
2. Corey a besoin D'UNE BIERE.
3. Fiona téléphone A RITA.
4. Joe-Bob sort AVEC TEX.
5. LE PROF dérange Joe-Bob.
6. LES DEVOIRS dérangent Joe-Bob.
7. Paw-Paw fait UNE SIESTE (nap).
8. Tammy cherche TEX.
9. TAMMY cherche Tex.
```

Figure 1: Grammar exercises from the *Tex's French Grammar* textbook

S: John hopes that Peter is liked by Mary.

To control the syntax and the lexicon of the exercices produced, we take a grammar based approach and make use of generation techniques. More specifically, we generate sentences using a Feature-Based Lexicalised Tree Adjoining Grammar (FB-LTAG) for French (SemTAG). We show that the rich linguistic information associated with sentences by the generation process naturally supports the identification of sentence pairs related by a syntactic transformation. In particular, we argue that the derivation trees of the FB-LTAG grammar provide a level of representation that captures both the formal and the content constraints governing transformations. The content words and the grammatical functions labelling the tree nodes permit checking that the two sentences stand in the appropriate semantic relation (i.e., fully identical content or identical content modulo some local change). Further, the syntactic properties labelling these nodes (names of FB-LTAG elementary tree names but also some additional information provided by our generator) permits ensuring that they stand in the appropriate syntactic relation.

The structure of the paper is the following. We start (Section 2) by discussing related work focusing on studies that target the production of syntactic reformulations. We then go on to present our approach and show that it permits generating different types of transformations (Section 3). In Section (4), we discuss results concerning linguistic coverage, precision and recall. Section (5) concludes with pointers for further research.

2 Related work

In linguistics, transformations (Harris, 1957; Chomsky, 1957) model recurrent linguistic relations between sentence pairs. For instance, a transformation can be used to define the relation between the active and the passive voice version of the same sentence. Formally, transformations were stated as tree-transducers on phrase structure trees and they defined either structure changing or structure building (generalised transformation) operations.

In computational linguistics, transformations and more generally, structure changing and structure building rules have been used in such tasks as text simplification (Siddharthan, 2010), text summarising (Cohn and Lapata, 2009) and question generation (Piwek and Boyer, 2012). In these approaches however, the transformation relation is not necessarily defined on phrase structure trees. For instance, for the question generation task, (Yao et al., 2012) has argued that Assertion/WH-Question transformations are best defined on semantic representations. Conversely, for text simplification, (Siddharthan, 2010) has convincingly shown that dependency trees are better suited as a representation on which to define text simplification rules than both phrase structure trees and semantic representations.

(Siddharthan, 2011) presents a user evaluation comparing different re-generation approaches for sentence simplification. He notes in particular that annotators preferred those transformations that are closer in syntax to the original sentence. To achieve this, rules for word ordering are either added to the transform rules or coded as constraints within the input to a generator. In contrast, in our approach, syntactic similarity can be deduced by tree comparison using the rich linguistic information associated by the generator with the FB-LTAG derivation trees.

(Chandrasekar and Srinivas, 1997) describes an algorithm by which generalised rules for simplification are automatically induced from annotated training material. Similar to our work, their approach makes use of TAG derivation trees as a

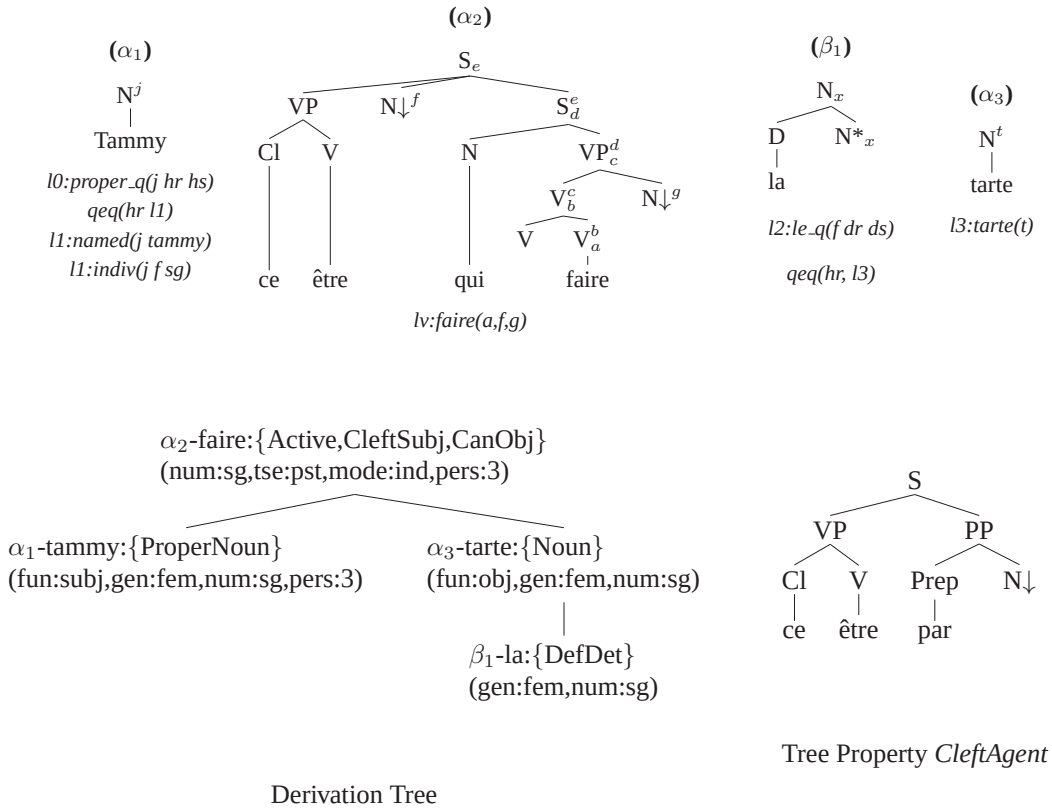


Figure 2: Grammar, Derivation Tree and Example Tree Property (Bottom right) for the sentence *C'est Tammy qui fait la tarte* (It is Tammy who bakes the pie)

base representation. Using a corpus of complex sentences parsed and aligned with the corresponding simplified sentences, the tree comparison algorithm they propose permit inducing simplification rules between dependency trees derived from TAG derivation trees. Although similar to our approach, (Chandrasekar and Srinivas, 1997)'s proposal differs from ours in several ways. First, while we focus on transformations, they work on simplifications relating e.g., a sentence containing a relative clause to two base clauses. Second, the trees on which they define their transformations are reconstructed in a rather ad hoc manner from the TAG derivation trees and from information extracted from the TAG derived trees. In contrast, we make use of the derivation trees produced by the *GraDe* algorithm. Third, while their work is limited to sentences containing relative clauses, we consider a wider range of transformations. Fourth, their approach targets the automatic acquisition of simplification rules while we manually define those.

3 Generating Transformation-related sentences

To generate pairs of sentences that are related by a transformation, we proceed in two main steps.

First, we construct a generation bank by generating sentences from underspecified semantic representations using the *GraDe* algorithm (Gardent and Kruszewski, 2012). This generation bank stores sentences that have been generated using *GraDe* together with the detailed linguistic information associated by this algorithm with each sentence in particular, its derivation tree.

Second, filters are used to retrieve from the generation bank sentence pairs that provide the query and the solution to a given transformation type exercise. These filters are defined on derivation trees and make use of the rich linguistic information associated by our generator with those derivation trees.

In what follows, we start by describing the grammar used and the information contained in the derivation trees produced by *GraDe* . We then go on to motivate the use of derivation trees as a structure on which to base the identification of

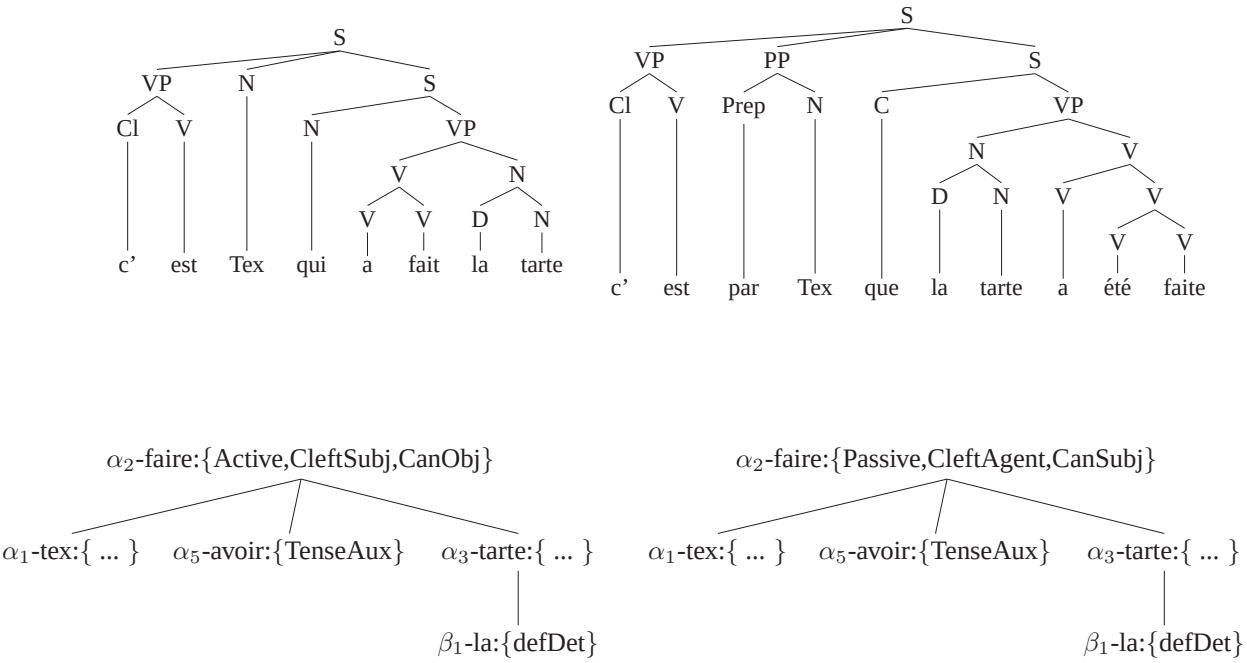


Figure 3: Derived (top) and Derivation (bottom) Trees for the active voiced sentence *C'est Tex qui a fait la tarte* (It is Tex who baked the pie) and its passive variant

transformationally related sentences. Finally, we present the derivation tree filters used to identify pairs of transformationally related sentences.

3.1 Grammar

The grammar used by the surface realiser is called *SemTAG*. It is a Feature-Based Lexicalised Tree Adjoining Grammar (FB-LTAG, (Vijay-Shanker and Joshi, 1988)) for French augmented with a unification-based compositional semantics as described in (Gardent and Kallmeyer, 2003).

Figure 2 shows an example FB-LTAG grammar and the derivation tree associated with the sentence *C'est Tammy qui fait la tarte* (It is Tammy who bakes a pie).

The basic elements of FB-LTAG are called *elementary trees*. Each elementary tree is labelled with feature structures and is associated with at least one lexical item called the *anchor* of that tree. Elementary trees are of two types: auxiliary (to model recursion) and initial (to capture predicate/argument dependencies). They are combined using two operations, *substitution* and *adjunction*. The result of combining elementary trees together is both a derived tree (representing phrase structure) and a derivation tree (describing the process by which a derived tree was produced). More specifically, an FB-LTAG derivation tree indicates which FB-LTAG elementary trees were used to construct the parse tree and how they were combined: each node in a derivation tree is labelled with the name of an elementary trees used in the derivation and each edge indicates which operation (substitution or adjunction) was applied to combine the two trees related by the edge.

As shown in Figure 2, the derivation trees produced by *GraDe* contain additional information[2]. Nodes are labelled not only with the name of an elementary tree but also with the lemma anchoring that tree, the feature structure associated with the anchor of that tree and the *tree properties* of that tree.

We use feature structure information to identify the grammatical function of an argument and to verify that two transformationally related sentences are syntactically and morpho-syntactically identical up to the transformed part.

Tree properties are abstractions over tree descriptions. These properties are produced by the grammar compiler computing the grammar out of a more abstract grammar specification. They name the tree descriptions that were used to build

[2]In Figure 2, edge labels are omitted for simplicity.

the FB-LTAG elementary trees. Thus, for instance, the tree property *CleftAgent* names the tree description appearing at the bottom right of Figure 2; and the elementary tree α_2 in Figure 2 is associated with the tree properties *Active, CleftSubj,CanObj* indicating that this tree was built by combining together the tree descriptions named *Active, CleftSubj* and *CanObj*.

3.2 Why Derivation Trees?

As discussed in Section 2, previous work on syntactic transformations has experimented with different levels of representation on which to define transformations namely, dependency trees, phrase structure tree and semantic representations. While providing detailed information about the syntax and the informational content of a sentence, FB-LTAG derivation trees provide both a more abstract description of this information than derived trees and a richer representation than semantic formulae.

Figure 3 illustrates the difference between derived and derivation trees by showing those trees for the sentences *C'est Tex qui a fait la tarte (It is Tex who baked the pie)* and *C'est par Tex que la tarte a été faite (It is by Tex that the pie was baked)*. While the derived trees of these two sentences differ in their overall structure (different structure, different number of nodes), their derivation trees are identical up to the tree properties of the verb. Moreover, the tree properties of the active ({*Active,CleftSubj,CanObj*}) and of the passive ({*passive,cleftAgent,canSubj*}) verb capture the changes in argument and verb realisation typical of a passive transformation. In other words, derivation trees provide a level of description that is simpler (less nodes) and that better supports the identification of tranformationally related sentences (more similar configurations and explicit description of changes in argument and verb realisation).

Derivation trees are also better suited than semantic formulae to capture transformations as, in some cases[3], the semantic representations of two transformationally related sentences may be identical. For instance, in our grammar, Active/Passive, canonical/inverted subject and cleft/non cleft argument variations are assigned the same semantics. As shown above, for those cases, the tree properties labelling the derivation trees provide a direct handle for identifying sentences related by these transformations.

3.3 Derivation Tree Filters

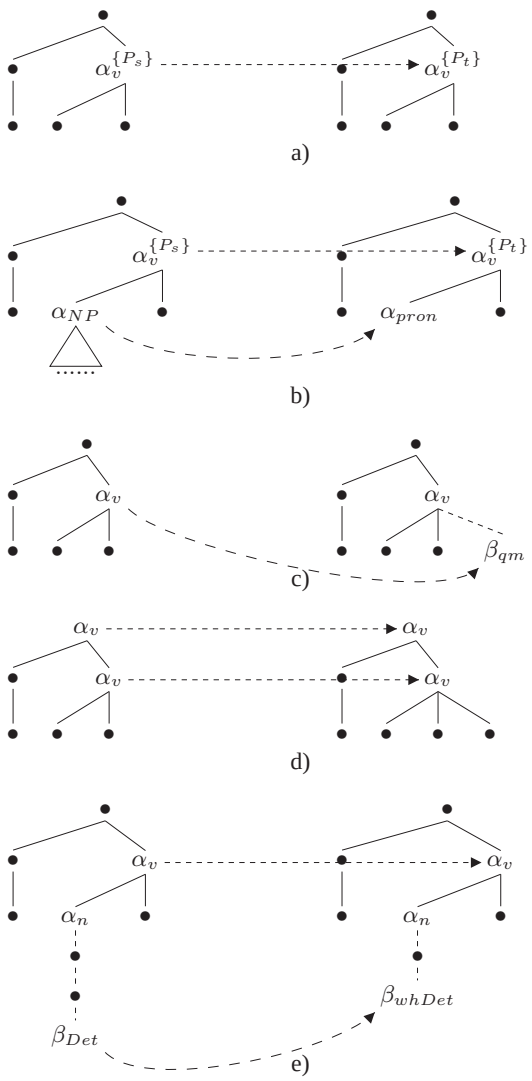


Figure 4: Tree filter types (tree schemas on the left depict source sentence derivation trees and those to their right their transform)

To identify transformationally related sentences, we define tree filters on derivation trees. These filters make use of all the information present in the FB-LTAG derivation trees produced by *GraDe* namely, the tree names, the lemmas, the feature structures and the tree properties la-

[3]Whether two syntactically distinct sentences share the same semantics depends on the grammar. In the grammar we use, the semantic representations aims to capture the truth conditions of a sentence not their pragmatic or informational content. As a result, Passive/Active variations do share the same semantics.

belling the nodes of these trees.

Figure 4 shows the general filtering patterns we used to handle four types of transformations used in language learning: Active/Passive, NP/pronoun (pronominalisation), NP/Wh-NP (WH-questions) and Assertion/Yes-No questions.

Filters (a) and (d) are used for the Active/Passive and for the canonical/inverted subject variations. Filter (a) relates two trees which are identical up to either one node differing in its tree properties. It applies for instance to the derivation trees shown in Figure 3. Filter (d) is used for cases such as *John wants Mary to like him / John wants to be liked by Mary* where the two derivation trees differ both in the tree properties assigned to *want* (*CanSubj, CanObj, SentObj* $\leftrightarrow$ *CanObj, SentObj*) and in the tree properties assigned to *like* (*InfSubj, CanObj* $\leftrightarrow$ *InfSubj, CanAgent*); and where an additional node is present due to the presence of the pronoun *him* in the active sentence and its absence in the passive variant.

Filter (b) is used for the NP/Pronoun transformation and relates two trees which in addition to having one node with different tree properties also differ in that an NP node and its subtree maps to a pronoun node.

Filter (c) relates two trees which are identical up to the addition of an auxiliary tree of type β_{qm}. As we shall see below, this is used to account for the relation between an assertion and a question including a question phrase (i.e., *n'est ce pas / Isn't it, est ce que*, inverted *t'il* or question mark).

Finally, Filter (e) is used for the assertion/wh-question transformation and matches pairs of trees such that an NP containing n modifiers in one tree becomes a WH-NP with any number of these n modifiers in the other tree.

We now discuss in more detail the derivation tree filters specified for each type of transformations.

3.4 Meaning Preserving Transformations

In SemTAG, semantic representations aim to capture the truth conditions of a sentence not their pragmatic or informational content. As a result, some sentences with different syntax share the same semantics. For instance, all sentences in (2b) share the semantics in (2a).

(2) a. L0:proper_q(C HR HS) L1:named(C tammy) L1:indiv(C f sg) qeq(HR L1) L3:love(EL TX C) L3:event(EL pst indet ind) L4:proper_q(TX HRX HSX) L5:named(TX tex) L5:indiv(TX m sg) qeq(HRX L5)

b. *Tex loves Tammy, It is Tex who loves Tammy, It is Tammy whom Tex loves, Tammy is loved by Tex, It is Tammy who is loved by Tex, It is by Tex that Tammy is loved, etc.*

The syntactic and pragmatic differences between these semantically identical sentences is captured by their derivation trees and in particular, by the tree properties labelling the nodes of these derivation trees. More generally, Active/Passive sentence pairs, canonical/cleft (e.g., *Tex loves Tammy / It is Tex who loves Tammy* and Canonical/Inverted Subject variations (e.g. *C'est Tex que Tammy adore / C'est Tex qu'adore Tammy* may lead to derivation trees of identical structure but distinct tree properties. In such cases, the transformationally related sentence pairs can therefore be captured using the first type of derivation filter i.e., filters which related derivation trees with identical structure but distinct tree properties. Here, we focus on the Active/Passive variation.

The differences between an active voice sentence and its passive counterpart include lexical, morphological and syntactic differences. Thus for instance, (3a) differs from (3b) in that the verb agrees with the proper name *Tammy* rather than the pronoun *il*; the clitic is in the oblique case (*lui*) rather that the nominative (*il*); the subject NP *Il* has become a PP headed by the preposition *par*; the passive auxiliary *être* and the preposition *par* have been added to support the passive voice construction.

(3) a. Il regarde Tammy (*He watches Tammy*)
b. Tammy est regardée par lui
(*Tammy is watched by him)*

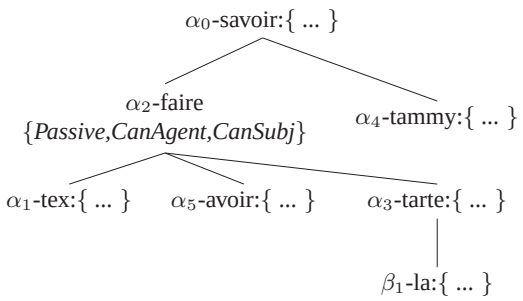
In (Siddharthan, 2010), these variations are handled by complex node deletion, lexical substitution, insertion, and node ordering rules. By contrast, to identify `Active / Passive` variations, we search for pairs of derivation trees that are related by an Active/Passive derivation tree filter namely, a filter that relates two trees which are identical up to a set of tree properties labelling a single node pair. We specify as many Active/Passive tree property patters as there are possible variations in argument realisations. For instance, for a transitive verb, some of the defined tree property patterns are as follows:

Active/Passive Tree Property Patterns

{*Active, CanSubj, CanObj*}
↔ {*Passive, CanSubj, CanAgent*}
{*Active, CliticSubj, CanObj*}
↔ {*Passive, CanSubj, CanAgent*}
{*Active, WhSubj, CanObj*}
↔ {*Passive, InvertedSubj, WhAgent*}
{*Active, RelSubj, CanObj*}
↔ {*Passive, CanSubj, RelAgent*}
{*Active, CleftSubj, CanObj*}
↔ {*Passive, CanSubj, CleftAgent*}

In sum, in our approach, the possible differences in morphological agreement between active and passive sentences are accounted for by the grammar; differences in argument realisation (Object/Subject, Subject/Agent) are handled by the tree filters; and lexical differences due to additional function words fall out of the FB-LTAG coanchoring mechanism.

As should be clear from the derivation tree below, our approach supports transformations at any level of embedding. For instance, it permits identifying the pair *Tammy sait que Tex a fait la tarte / Tammy sait que la tarte a été faite par Tex (Tammy knows that Tex has baked the pie / Tammy knows that the pie has been baked by Tex)*.



It also supports a fine grained control of the Active/passive variants allowing both for cases with multiple variants (4a) and for transitive configurations with no passive counterpart (4b,d).

(4) a. C'est la tatou qu'il adore
C'est par lui que la tatou est adorée
C'est lui par qui la tatou est adorée
It is the tatoo that he loves / It is the tatoo that is loved by him

b. Tex veut faire une tarte
** Une tarte veut être faite par Tex
*Tex wants to bake a pie / ** A pie wants to be baked by Tex*

c. Tex semble faire une tarte
Une tarte semble être faite par Tex
Tex seems to bake a pie / A pie seems to be baked by Tex

d. Tex mesure 1.80m
** 1.80m est mesuré par Tex
*Tex measures 1.80m ** 1.80m is measured by Tex*

(4a) is accounted for by specifying a tree filter including the tree property mapping *CleftSubject* ↔ *CleftAgent* where *CleftAgent* subsumes the two types of clefts illustrated in (4a).

The lack of passive in (4b) and (4d) is accounted for by the grammar: since (4b) does not licence a passive, the starred sentence will not be generated. Similarly, because the verb *mesurer / to be X tall* is not passivable, the starred sentence in (4d) will not be produced.

3.5 Meaning Altering Transformations

When the content of two sentences differs, in particular, when a content word is deleted or added, the derivation trees of these sentences may have a different structure. In those cases, we use filters that relate derivation trees with distinct tree structures namely, filters (b), (c), (d) and (e) in Figure 4.

NP/Pronoun To handle the NP/Pronoun, we use the filter sketched in Figure (4b) which relates derivation trees that are identical up to an NP subtree replaced by a node labelled with a pronoun. In this way the difference between the derivation tree of *le tatou* (two nodes) and *qui* (one node) does not prevent the identification of sentence pairs such as (5a).

(5) a. Le tatou chante
Il chante (*Personal pronoun*)
The tatoo sings/He sings

b. Quel tatou chante ?
Qui chante ? (*WH-Personal Pronoun*)
Which tatoo sings?/Who sings?

NP/Wh-NP For wh-questions, the main difficulty is to account for variations such as (6) below where a complex NP with several modifiers can map to a Wh-NP with different numbers of modifiers. To capture these various cases, we use two tree filters. The first filter is similar to filter

(b) in Figure 4 and matches NP/WH-Pronoun sentences (e.g., 6a-b where the NP *Le grand tatou avec un chapeau qui dort sous le palmier* maps to a WH-Pronoun *qui*). The second tree filter is sketched in Figure (4e). It matches NP/Wh-NP sentences (e.g., 6a-c/f) where an NP matches to a WH-NP headed by a WH-Determiner, the head noun and any number of modifiers.

(6) a. Le grand tatou avec un chapeau qui dort sous le palmier ronfle.
Qui ronfle ? Quel tatou ronfle ? Quel grand tatou ronfle? Quel tatou avec un chapeau ronfle ? Quel tatou qui dort sous un palmier ronfle ? etc.
The big tatoo with a hat who sleeps under the palmtree snores/ Who snores? Which tatoo snores? Which tatoo with a hat snores? Which tatoo who sleeps snores? etc.

Yes-No Question. In French, yes/no questions can be formed in several ways:

(7) a. Le tatou chante
Le tatou chante t'il? (*Inverted t'il*)
Est ce que le tatou chante ? (*est ce que*)
Le tatou chante? (*Intonation*)
Le tatou chante n'est ce pas? (*n'est ce pas (isn't it)*)
The tatoo sings / Does the tatoo sing? The tatoo sings? The tatoo sings doesn't it?

b. Vous chantez
Chantez vous? (*Inverted Subject*)
You sing/Do you sing?

For cases such as (7b), we require the derivation trees to be identical up to the tree property mapping *CliticSubject* $\leftrightarrow$ *InvertedCliticSubject*. For cases such as (7a) on the other hand, we use the filter sketched in Figure (4c) that is, a filter which requires that the derivation trees be identical up to a single additional node licenced by a question phase (i.e., *t'il, est ce que, n'est ce pas* or a question mark).

4 Evaluation

We carried out an experiment designed to assess the genericity, the correctness, the coverage and the recall of the approach. In what follows, we describe the grammar and lexicon used in that experiment; the sentence set used for testing; and the results obtained.

Grammar and Lexicons. The *SemTAG* grammar used contains around 1300 elementary trees and covers auxiliaries, copula, raising and small clause constructions, relative clauses, infinitives, gerunds, passives, adjuncts, wh-clefts, PRO constructions, imperatives and 15 distinct subcategorisation frames. The syntactic and morphosyntactic lexicons used for generating were tailored to cover basic vocabulary as defined by the lexicon used in *Tex's French Grammar*. The syntactic lexicon contains 690 lemmas and the morphological lexicon 5294 forms.

Generated Sentences. To populate the generation bank, we input *GraDe* with 52 semantic formulae corresponding to various syntactic and semantic configurations and their interactions[4]: including all types of realisations for verb arguments (cleft, pronominalisation, relative, question arguments); Intransitive, Transitive and ditransitive verbs; Control, raising and embedding verbs; Nouns, common nouns, personal strong and weak pronouns; standard and Wh- determiners.

From these 52 semantic formulae, *GraDe* produced 5748 sentences which we stored in a database together with their full semantics and their derivation tree.

Results. Table 1 summarises the results of our experiment. It indicates the number of source sentences manually selected so as to test different syntactic configurations for each type of transformation considered (S), the number of transformations found for these source sentences (T), the number of tree filters used for each type of transformation (TF) and the precision obtained (ratio of correct transformations).

The low number of tree filters relative to the number of syntactic configurations explored indicates a good level of genericity: with few filters, a transformation can be captured in many distinct syntactic contexts. For instance, for the Active/passive transformation, 8 filters suffice to capture 43 distinct syntactic configurations.

As expected in an approach where the filters are defined manually, precision is high indicating that the filters are accurate. The generated pairs marked as incorrect by the annotator are all cases where the transformed sentence was ungrammatical; in other words, the filters were accurate.

[4]We restrict the tense of the verb of the main clause to present and indicative mode

	S	T	TF	Precision
Active/passive	43	38	8	88.5
Pronominalisation	36	33	7	73
Wh-Questions	25	20	7	88
YN-Questions	24	20	5	90.5

Table 1: Source Sentences (S), Transformations of Source Sentences (T), Number of Filters (F) and Precision (Ratio of correct transformations)

Finally, the relatively low number of transformations found relative to the number of source sentences (e.g., 38 transforms for 43 source sentences in the Active/passive case) is mainly due to transformed sentences that are missing from the generation bank either because the corresponding input semantics is missing or because of gaps in the grammar or the lexicon. However, for few cases missing filters were identified as well.

5 Conclusion

We presented an approach which is, to the best of our knowledge, the first approach for generating grammar exercices covering a wide range of structure changing transformations. And we argued that FB-LTAG derivation trees naturally support the identification of sentences that are related by a syntactic transformation.

In current work, we are pursuing two main directions. First, we are investigating how to account for more complex transformations such as *Tom ate because of his hunger* and *His hunger caused Tom to eat*. In particular, we plan to explore in how far the approach developed on dependency trees by (Siddharthan, 2010) can be ported to SemTAG derivation trees. Second, drawing on (Chandrasekar and Srinivas, 1997)'s work, we are investigating how to develop an algorithm that can induce derivation tree filters from FB-LTAG derivation trees.

Acknowledgments

The research presented in this paper was partially supported by the European Fund for Regional Development within the framework of the INTERREG IV A Allegro Project. We would also like to thank German Kruszewski and Elise Fetet for their help in developing and annotating the test data.

References

I. Aldabe, M. Lopez de Lacalle, M. Maritxalar, E. Martinez, and L. Uria. 2006. Arikiturri: an automatic question generator based on corpora and nlp techniques. In *Proceedings of the 8th international conference on Intelligent Tutoring Systems*, ITS'06, pages 584–594, Berlin, Heidelberg. Springer-Verlag.

R. Chandrasekar and B. Srinivas. 1997. Automatic induction of rules for text simplification. *Knowledge-Based Systems*, pages 183–190.

C. Chen, H. Liou, and J. Chang. 2006. Fast: an automatic generation system for grammar tests. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 1–4, Stroudsburg, PA, USA.

N. Chomsky. 1957. *Syntactic Structures*. Mouton.

T. Cohn and M. Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674.

C. Gardent and L. Kallmeyer. 2003. Semantic construction in FTAG. In *10th EACL*, Budapest, Hungary.

C. Gardent and G. Kruszewski. 2012. Generation for grammar engineering. In *INLG 2012, The seventh International Natural Language Generation Conference*, Starved Rock, Illinois, USA.

Z.S. Harris. 1957. Co-occurrence and transformation in linguistic structure. *Language*, 33(3):283–340.

L. Perez-Beltrachini, C. Gardent, and G. Kruszewski. 2012. Generating Grammar Exercises. In *NAACL-HLT 7th Workshop on Innovative Use of NLP for Building Educational Applications*, Montreal, Canada.

P. Piwek and K. E. Boyer. 2012. Varieties of question generation: introduction to this special issue. *Dialogue and Discourse, Special Issue on Question Generation*.

A. Siddharthan. 2010. Complex lexico-syntactic reformulation of sentences using typed dependency representations. In *Proceedings of the 6th International Natural Language Generation Conference*, INLG '10, pages 125–133, Stroudsburg, PA, USA.

A. Siddharthan. 2011. Text simplification using typed dependencies: a comparison of the robustness of different generation strategies. In *Proceedings of the 13th European Workshop on Natural Language Generation*, ENLG '11, pages 2–11, Stroudsburg, PA, USA.

K. Vijay-Shanker and AK Joshi. 1988. Feature Structures Based Tree Adjoining Grammars. *Proceedings of the 12th conference on Computational linguistics*.

X. Yao, G. Bouma, and Y. Zhang. 2012. Semantics-based question generation and implementation. *Dialogue and Discourse, Special Issue on Question Generation*.