GRASS: A SYNTACTIC TEXT SIMPLIFICATION SYSTEM BASED ON SEMANTIC REPRESENTATIONS

Rita Hijazi, Bernard Espinasse, Núria Gala

GRASS: A SYNTACTIC TEXT SIMPLIFICATION SYSTEM BASED ON SEMANTIC REPRESENTATIONS

Rita Hijazi[1,2], Bernard Espinasse[1] and Núria Gala[2]

[1] Aix-Marseille Univ., Laboratoire Informatique et Systèmes (LIS UMR 7020), Marseille, France
[2] Aix-Marseille Univ., Laboratoire Parole et Langage (LPL UMR 7309), Aix-en-Provence, France
{rita.hijazi,bernard.espinasse,nuria.gala}@univ-amu.fr

ABSTRACT

Automatic Text Simplification (ATS) is the process of reducing a text's linguistic complexity to improve its understandability and readability while maintaining its original information, content, and meaning. Several text transformation operations can be performed such as splitting a sentence into several shorter sentences, substitution of complex elements, and reorganization. It has been shown that the implementation of these operations essentially at a syntactic level causes several problems that could be solved by using semantic representations. In this paper, we present GRASS (GRAph-based Semantic representation for syntactic Simplification), a rule-based automatic syntactic simplification system that uses semantic representations. The system allows the syntactic transformation of complex constructions, such as subordination clauses, appositive clauses, coordination clauses, and passive forms into simpler sentences. It is based on graph-based meaning representation of the text expressed in DMRS (Dependency Minimal Recursion Semantics) notation and it uses rewriting rules. The experimental results obtained on a reference corpus and according to specific metrics outperform the results obtained by other state of the art systems on the same reference corpus.

KEYWORDS

Syntactic Text Simplification, Graph-Based Meaning Representation, DMRS, Graph-Rewriting

1. INTRODUCTION

Automatic Text Simplification (ATS) transforms a complex text into an equivalent version that would be easier to read and/or understand by a target audience without significantly changing the input original meaning [1]. Simplification has been shown useful both as a pre-processing step for Natural Language Processing (NLP) tasks such as machine translation [2], relation extraction [3], text summarization [4], and for developing reading aids, e.g., for people with dyslexia [5], individuals with low vision [6], or non-native speakers [7]. Traditionally, two different tasks are considered in ATS: *lexical simplification* and *syntactic simplification*. Roughly speaking, *lexical simplification* (LS) consists of complex word identification and substitution by a simpler synonym or adding definitions. *Syntactic simplification* (SS) aims to transform sentences containing syntactic constructions that may hinder readability and comprehension into more readable or understandable equivalents. Several text transformation operations can be performed such as *division*, consisting of splitting a sentence into multiple shorter sentences, *deletion*, *reorganization*, and *morpho-syntactic substitutions*.

In this paper, we present GRASS (GRAph-based Semantic representation for syntactic Simplification), an automatic syntactic simplification system, and we focus on sentence splitting

and passive to active voice transformations using graphs as semantic representations[1]. GRASS implements a specific syntactic simplification method based on rewriting rules that exploit a semantic representation. This semantic representation of the text is expressed in Dependency Minimal Recursion Semantics notation (DMRS) [8]. Both semantic and syntactic information are expressed in the text, which simplifies the splitting operation. The simplification process in GRASS is done according three steps: (i) semantic representation of the complex sentence by a DMRS graph; (ii) transformation of this DMRS graph into one or several DMRS graphs by applying a set of transformation rules; and (ii) generation of simplified sentence(s) from the transformed DMRS graph(s).

GRASS system is automatically evaluated on the HSplit corpus [9] according to a set of reference metrics (BLEU, SARI, SAMSA) used in automatic text simplification. We compare the results obtained with GRASS with two state-of-the-art syntactic semantic-based simplification systems, HYBRID [10] and DSS [11]. We show that our system outperforms both HYBRID and DSS in syntactic simplification of the targeted structures.

The paper is organized as follows: section 2 introduces ATS main current approaches, with a special focus on semantic-based ATS systems. Section 3 presents GRASS, its theoretical foundations, and its software architecture. The experimental setup is detailed in section 4. Section 5 presents the results obtained by our tool, that we compare with the results obtained by other systems on the same reference corpus. We finally conclude with some perspectives of this work.

2. RELATED WORK

In this section we first present some mainstream approaches of automatic text simplification, and we then focus on semantic-based syntactic simplification.

2.1. Automatic Text Simplification

Text simplification mainly concerns two main linguistic levels of simplification: lexical and syntactic. To perform these simplifications, three main approaches can be identified: *rule-based approaches*, *machine learning-based approaches*, and a combination of both, known as *hybrid approaches*.

Rule-based approaches were the first to appear. Concerning syntactic simplification, specific hand-crafted sentence splitting rules were first proposed by [12] and [13]. Rule-based approaches are generally used for specific applications and for a well-targeted populations [14][15]. They rely on a study of corpora to identify linguistic phenomena affecting readability or comprehensibility. The idea here is to isolate a set of complex structures, and to create transformation rules to paraphrase. According to [16], manual rules are used in the field of text simplification when a system focuses on very specific linguistic structures and phenomena that are relatively easy to manage with a limited set of rules. However, their compilation and validation are laborious [17], i.e., they require expert human involvement and lead to linguistically accurate simplification systems.

In many cases, syntax transformation rules are implemented using synchronous grammars [18], which specify transformation operations between syntax trees using many rules. For example, [19] used 111 rules for appositions, subordination, coordination, and relative clauses. [20] presented a rule-based system to automatically simplify Brazilian Portuguese text for people with low literacy. They proposed a set of operations to simplify 22 syntactic constructions. [14] followed a similar approach for French syntactic simplification, using manually constructed rules based on a typology of simplification rules manually extracted from a corpus of simplified

[1] The system code and results can be found on GitHub: https://github.com/RitaHijazi/Semantic-based-Text-Simplification

French. [21] described a simplification of Spanish text that can simplify relatives, coordination, and participles. These rule-based systems often face several problems when dealing with long sentences, e.g., identifying the splitting points, rewriting shared elements, and deleting verb arguments which are needed for comprehension [10].

Machine Learning-based approaches, also called *corpus-based approaches*, have more recently been proposed in search of more robustness and coverage and to reduce the human involvement of the previous approach. The ATS systems developed based on these approaches generally use deep learning techniques (neural networks and word embeddings) and exploit large parallel corpora, i.e., original texts having simpler variants, e.g., Newsela [22] [23] and Wikipedia-Simple English Wikipedia [24] [25].

These approaches mainly consider the simplification task as a monolingual variant of a machine translation (MT) task. However, most of the simplified sentences are very similar to the complex sentence, and as such they are not suitable for the evaluation of full-fledged sentence simplification systems performing more complex sentence splitting and rewriting operations. That's why these models do not address sentence splitting.

The ATS systems developed according to this approach are generally efficient for lexical simplification but still present important limitations for syntactic simplification. The main drawback of these approaches is that the simplifications are not straightforwardly interpretable to humans (these models are often called 'black boxes') which can undermine trust in those models when it comes to evaluation of the results (i.e., when parallel corpora are not big enough).

Hybrid approaches try to take advantage of the benefits of the two previous approaches, mostly by combining rule-based syntactic simplifications, and lexical simplifications with learning-based approaches [10][11][26]. However, in this combination, to resolve limitations of rule-based systems for syntactic simplification, syntactic structures do not always capture the semantic arguments of a frame, which may result in wrong splitting boundaries [10]. To solve this problem, the authors working on hybrid approaches have proposed to take advantage of the semantic structures for sentence division.

2.2. Semantic-Based Syntactic Simplification

To our knowledge, [10] [26] are the first to propose to use semantic structures for sentence division in syntactic simplification. The operations of division and deletion are driven by semantics: the division is determined by the semantic roles that are associated with an element while the deletion of a node is determined by its semantic relationships with the divided events. Hence, their deletion model distinguishes between arguments and modifiers using a small number of rules. [10] proposed HYBRID, a supervised system that uses semantic structures, the Discourse Representation Structure [28] for sentence splitting and deletion. Splitting candidates are pairs of event variables associated with at least one core thematic role (e.g., agent or patient). Semantic annotation is used on the source side in both training and test of the system.

A little later, [26] proposed an unsupervised pipeline, where sentences are split based on a probabilistic model trained on the semantic structures of Simple Wikipedia, as well as a language model trained on the same corpus. [29] proposed the Split and Rephrase task, focusing on sentence splitting. For this purpose, they presented a specialized parallel corpus, derived from the WebNLG dataset [30]. The latter is obtained from the DBpedia knowledge base [31] using content selection and crowdsourcing. It is annotated with semantic triplets of subject-relation-object, obtained semi-automatically.

More recently, [11] have combined structural semantics with rules for syntactic simplification and neural methods for lexical simplification. They presented Direct Semantic Splitting (DSS), an algorithm (based on rules) using a semantic parser which supports the direct decomposition

of the sentence into its main semantic constituents. They use the UCCA semantic notation for semantic representation of the sentence [32]. UCCA aims to represent the main semantic phenomena in the text, without taking into consideration the syntactic forms. After splitting, NMT-based simplification system [33] is performed for lexical simplification.

While taking into account semantics is paramount, a system that would be only based on semantics does not seem appropriate for syntactic simplification. The argument-predicate relation is not enough to detect all the syntactic structures, both semantic and syntactic information are needed. Our research adopts the same approach as [11], focusing on syntactic simplification. It is based on a rule-based approach, but it uses the DMRS notation, which unlike UCCA combines the semantic and the syntactic representation of a sentence.

3. THE GRASS SYSTEM

GRASS for GRAph-based Semantic representation for syntactic Simplification, is a rule-based automatic syntactic simplification system that uses semantic representations. It allows the syntactic transformation of complex sentences with syntactic constructions, such as subordination clauses, appositive clauses, coordination clauses and transformation from passive to active form into simpler constructions. As GRASS performs only syntactic simplification, as HYBRID and DSS systems do, it can be coupled with existing lexical simplification systems such as neural systems NTS [33].

In this section, we first present GRASS theoretical foundations, particularly the DMRS semantic graph representation and the DMRS-based simplification method. We then describe the GRASS software architecture with its components. We finally present a simplification example of appositive sentence transformed with GRASS.

3.1. GRASS Theoretical Foundations

GRASS uses the DMRS scheme for semantic representation [8]. DMRS differs from UCCA and DRS respectively used by [11] and [10] in the way the information is expressed. DMRS semantics are rooted in the superficial form of sentences and in the syntactic links between constituents. DMRS, as most semantic representations, rely on syntactic analyses: there is a strong overlap between semantic and syntactic constituents. DMRS semantics are anchored in the surface form of the sentences and in the syntactic links between the constituents. Syntactic information is explicitly marked, e.g., subordination, apposition, etc.

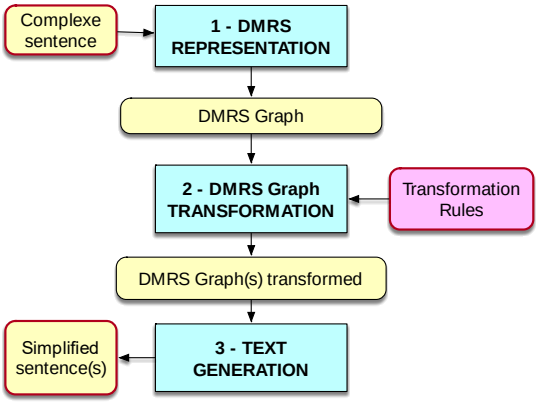


Figure 1. Steps of the syntactic simplification method.

GRASS implements a specific syntactic simplification method based on DMRS semantics and structured in three main steps as illustrated in Figure 1. The first one aims at representing the complex sentence by a DMRS graph-based meaning representation. The second step is to transform this DMRS graph into one or several DMRS graphs by applying a set of

transformation rules defined manually (simplification rules). The third step consists of generating the simplified sentences from these transformed DMRS graphs.

GRASS is based on the English Resource Grammar (ERG) [34], a broad-coverage, symbolic grammar of English, developed as part of DELPH-IN[2] initiative and LinGO[3] project. The ERG uses Minimal Recursion Semantics (MRS) [35] as semantic representation. The MRS format can be transformed into a more readable DMRS graph, which represents its dependency structure. The nodes correspond to predicates; edges, referred to as links, represent relations between them.

The ERG grammar is a bidirectional grammar which supports both parsing and generation. Several processors exist to parse sentences into MRSs and generate surface forms from MRS representations using chart generation. In our experiments, we used ACE[4] to obtain DMRSs graphs and to generate other graphs from them. Parsing and generation are thus performed using already existing DELPH-IN tools. DMRS has already been used in other systems for prepositional phrase attachment disambiguation [36], for machine translation [37], for question generation [38], for evaluating multimodal deep learning models [39], and for sentiment analysis [40].

The DMRS notation considers both semantic and syntactic annotations of sentences. This enables to detect the syntactic constructions that has to be transformed. The semantically shared elements are kept to be able to rewrite them into the split sentence. This allows to have a simpler output which is both grammatical (syntactic information from DMRS) and to preserve the meaning (information related to semantics in DMRS). DMRS provides information about the thematic roles which are necessary to reconstruct the shared elements, and to detect complex syntactic constructions.

DMRS graphs can be manipulated using two existing Python libraries. The pyDelphin[5] library is a more general MRS-dedicated library. It allows conversions between MRS and DMRS representations but internally performs operations on MRS objects.

We developed our simplification rules by examining data in raw texts and by transforming structural pattems into DMRS graphs. Currently, GRASS permits the syntactic simplification of 5 grammatical constructions: coordination (1), subordination (2), appositive clauses (3), relative clauses (4), passive forms (5). The DMRS representation of these sentences is showed in Figure 2. For the sake of clarity, we have modified the DMRS by deleting some elements in the sentences.

(1) The wave traveled across the Atlantic, and organized into a tropical depression off the northern coast of Haiti on September 13.

(2) He settled in London, devoting himself chiefly to practical teaching.

(3) Finally, in 1482, the Order dispatched him to Florence, the city of his destiny.

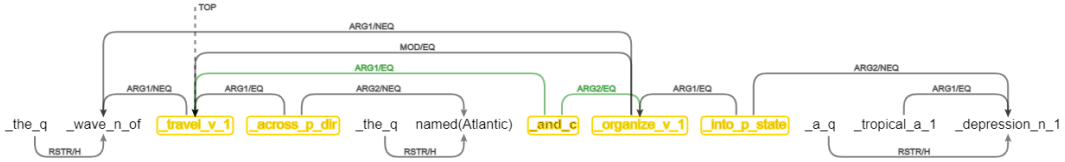
(4) It is located on an old portage trail which led west through the mountains to Unalakleet.

(5) Most of the songs were written by Richard M. Sherman and Robert B. Sherman.

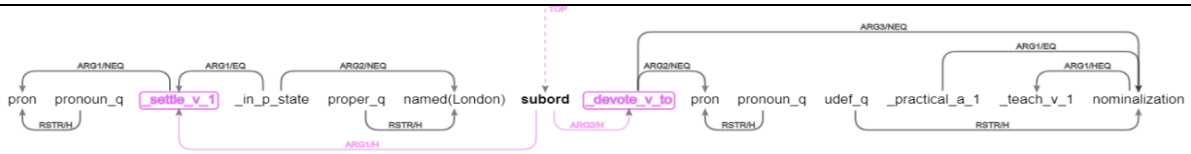
To simplify these constructions, we extract triggering indicators (the arguments of conjunctions or prepositions). For each segmentation, we identify a splitting point that acts as a trigger, i.e., its presence indicates the possibility of a segmentation. The development of the rules depends on the structure of the sentences in English. This involves studying each of the syntactic

[2] http://moin.delph-in.net/wiki/
[3] LINguistic Grammars Online, https://www-csli.stanford.edu/groups/lingo-project
[4] http://sweaglesw.org/linguistics/ace/
[5] https://github.com/delph-in/pydelphin

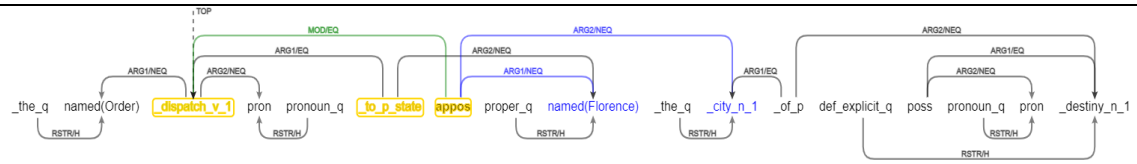
constructions to be processed, drawing up the "patterns" of constructions' forms and translating them into manual rules.



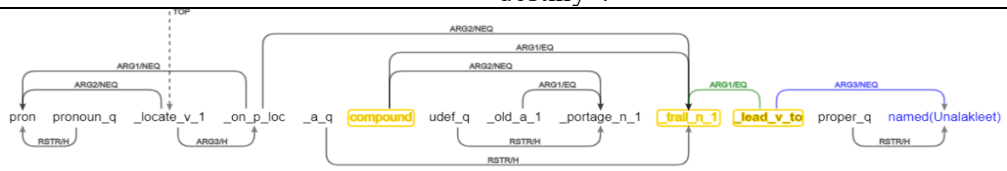
a. DMRS of sentence 1. "The wave traveled across the Atlantic, and organized into a tropical depression off the northern coast of Haiti on September 13".



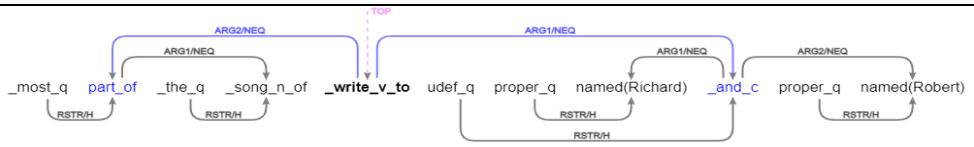
b. DMRS of sentence 2. "He settled in London, devoting himself chiefly to practical teaching".



c. DMRS of sentence 3. "Finally, in 1482, the Order dispatched him to Florence, the city of his destiny".



d. DMRS of sentence 4. "It is located on an old portage trail which led west through the mountains to Unalakleet".



e. DMRS of sentence 5. "Most of the songs were written by Richard M. Sherman and Robert B. Sherman".

Figure 2. DMRS graphs for sentences 1 to 5

3.2. GRASS Software Architecture

As illustrated in Figure 3, the software architecture is made of the following components: *Text Preparation*, *Semantic Parsing*, *Simplification* and *Text Generation*. In addition, there is a DMRS *graph visualization* component.

3.2.1. Preparation Component

This component prepares the corpus for simplification. The first operation is to put it in an interpretable format for the "Semantic Parsing" component (it transforms each sentence of the corpus into a DMRS semantic graph). In particular, the corpus to be processed must be divided

into sentences. It is important to preserve the position of the sentences in the original corpus to be able to generate them in the right place.

3.2.2. Semantic Parsing Component

Semantic parsing is performed by the ACE component, developed by the DELPH-IN Consortium. ACE is an efficient processor for DELPH-IN HPSG grammars: ACE allows both to translate a sentence into a DMRS graph (ACE parser) and to generate a sentence from a DMRS graph (ACE generator). A sentence is taken as input and the output is an associated MRS format file describing the semantic information. MRS format cannot be handled by the tools that we have chosen to use for visualization and transformation. Therefore, it has to be transformed into a DMRS graph using a DELPH-IN utility.

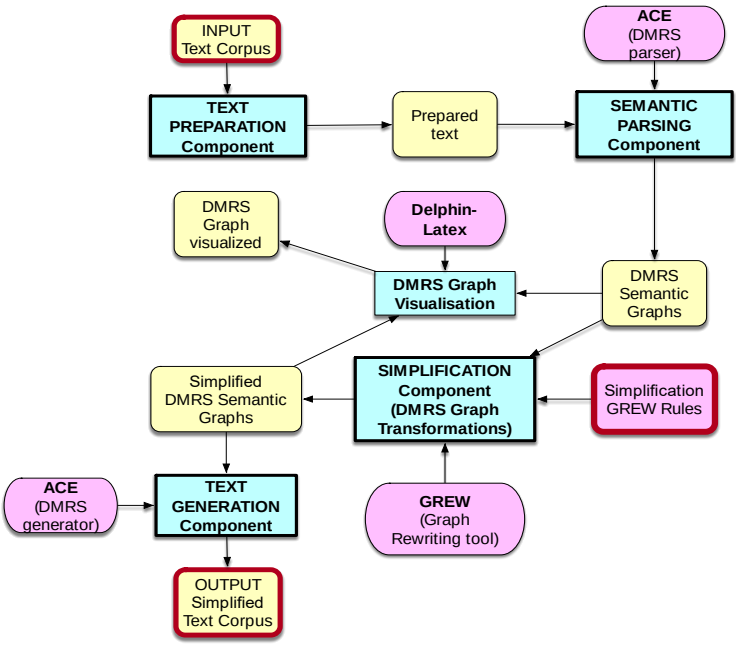


Figure 3. Software Architecture of GRASS system with its main components.

3.2.3. Simplification Component (DMRS Graphs Transformation)

This component simplifies the corpus, sentence by sentence, at the level of the DMRS graphs associated with each sentence of the corpus. It is based on GREW[6] [41] [42] [43] developed at the LORIA laboratory of INRIA.

GREW is a Graph REWriting tool for applications in NLP that can manipulate syntactic and semantic representations. It is used on POS-tagged sequences, surface dependency syntax analysis, deep dependency parsing, and semantic representation (AMR, DMRS). It can also be used to represent any graph-based structure. As such, GREW permits to transform graph-based semantic representations in DMRS according to a set of rules.

Hand-crafted rules can be defined and applied on a DMRS graph. The rules are structured into three sections: (i) *pattern*: describes the part of graph to match, allowing the selection of nodes or edges thanks to their features, relations or positions in the graph; (ii) *without*: filters out unwanted occurrences of the pattern giving the possibility to exclude elements from a previous selection; (iii) *commands*: allows to apply structural transformations on the graph, such as the deletion, the creation or the reordering of the nodes and edges as well as the modification of

[6] https://www.grew.fr/

their features in the graph. Each simplification operation transforming a DMRS graph is associated with a set of GREW rules (cf. section 3.3).

3.2.4. Generation Component

From the DMRS representations of the sentences of the corpus transformed by the GREW rules, this component generates the text associated with each sentence and places each generated sentence in the order of the original corpus. This component is based on the ACE tool that we already used for Semantic Parsing.

3.2.5. DMRS Graph Visualization Component

Delphin-Latex component, developed by the DELPH-IN Consortium [44], is a tool that takes as input a representation expressed in DMRS and visualizes the associated DMRS graph. This tool is very useful for the development of GREW simplification rules. It allows to visualize the DMRS representation before and after simplification.

3.3. Syntactic Simplification Rules

Our work enabled us to create simplification rules to transform DMRS graphs into other graphs. As regards to sentence splitting, we dealt with coordination, subordination, apposition, and relative clauses. We also worked on transformation from passive to active voices. These transformation rules, presented at an abstract level, are implemented in GREW. Our system contains 11 rules: 3 for apposition clauses, 3 for coordination clauses, 1 for passive to active voice transformation, 2 for relative clauses and 2 for subordination clauses. An example of GREW rule for rewrite one type of appositive clauses is presented in Figure 4.

3.3.1. Rules for Coordination Clauses

Coordination is formed by two or more elements linked by a conjunction such as "and", "or", etc. In DMRS, coordinations are identified by any relationship that has a *_c_* suffix, such as *_and_c_* and *_or_c_*. Coordination between propositions (not two nouns or adjectives) is our goal in splitting coordination. There are two types of coordinations between two clauses: clauses that share the same subject and clauses that do not share the same subject. We deal with these two cases. Sentence 1 is an example of coordination clause that sharing subject (*the wave*). The conjunction node C (*_and_c_*) takes the two verbs (*travel* V1 of the first clause and *organize* V2 of the second clause) of the two clauses as Arguments. The goal is to delete the conjunction C and to rewrite the shared subject (*the wave*) labeled ARG1/NEQ before the second verb adding edges between V2 and the rewritten subject. Sentence 1 can be transformed into two simpler sentences: *The wave traveled across the Atlantic. The wave organized into a tropical depression off the northern coast of Haiti on September 13.*

3.3.2. Rules for Subordination Clauses

In DMRS, subordination is marked by the label *_subord_*. The ARG1 of the subordinate clause refers to the main clause while the ARG2 refers to the subordinate clause (sentence 2). Thus, the splitting rule extracts all nodes linked to ARG1/2 separately and builds two new DMRSs. The goal is to transform a subordinate into a main and rewrite the shared subject. Sentence 2 can be transformed into two simpler sentences: *He settled in London. He devoted himself chiefly to practical teaching.*

3.3.3. Rules for Appositive Clauses

Apposition is formed by two adjacent nouns describing the same reference in a sentence. In DMRS, apposition in sentences can be captured precisely: it is identified by the label *appos* that takes the two adjacent nouns as arguments (sentence 3). The apposition splitting rule first

duplicates the ARG1 of the node *appos*, removes it to form the first DMRS, then it builds the other DMRS by replacing *appos'* ARG1 with its ARG2. The second step is to add the verb to be in present simple after the reproduced subject. The last step is to add links between the verb to be, the subject and the object. Sentence 3 can be transformed into two simpler sentences: *Finally, in 1482, the Order dispatched him to Florence. Florence is the city of his destiny*.

3.3.4. Rules for Relative Clauses

Although relative pronouns indicate relative clauses, in a DMRS structure these relative pronouns are not explicitly represented: there is not a node for the relative pronoun "that". However, the verb lead governs its subject by an /EQ relation. This indicates that *lead* and *trail* share the same tag and have the same scope. After splitting the sentence, this constraint of the same scope must be resolved. Sentence 4 can be transformed into two sentences: *It is located on an old portage trail. The trail led west through the mountains to Unalakleet*.

3.3.5. Rules for Transformation from Passive to Active Voices

A sentence in its active or passive form has two syntactic analyses, but the same semantic representation, hence the ease of the task by reversing the two arguments of the verb. In DMRS, the ARG1 the passive voice is the subject and ARG2 is the object. The goal is to reverse them to have ARG1 and ARG2 as object and subject respectively. Sentence 5 can be transformed into: *Richard M. Sherman and Robert B. Sherman wrote most of the songs.*

```
  rule appos {
    pattern { %pattern of the graph to transform node appos and
the arguments
  R [gpred="appos"];
  m: R -[ARG1:NEQ]-> N;
  p: R -[ARG2:NEQ]-> M;}

    commands  { %delete node appos, add verb to be, rewrite the
ARG1 node of the appos
  del_node R;
  add_node N1 :> N; append_feats N ==> N1;
  add_node V :>N ;
  V.lemma=be;V.pos="v";V.TENSE=pres; V.MOOD=indicative;
  add_edge V -[ARG2:NEQ]-> M;
  add_edge V -[ARG1:NEQ]-> N }}
```

Figure 4. Example of GREW rule for one case of appositive clause

4. EXPERIMENTAL SETUP

In this section we define the reference corpus and metrics used to evaluate GRASS.

4.1. Corpus

All systems including ours are tested on the HSplit[7], the test corpus of [9] (the authors highlight that existing English Wikipedia-based datasets did not contain sufficient instances of sentence splitting). To overcome this problem, they collected four reference simplifications of this kind of transformation for all 359 original sentences in the Turkcorpus test set [22]. TurkCorpus[8] comprises 359 sentences from the PWKP corpus [24] with 8 references collected by crowdsourcing for each of the sentences. In HSplit, each reference was created in only operating sentence splitting on the original complex sentence, so this is a data set for evaluating sentence splitting, but it does not generalize to sentence simplification in general.

[7] https://github.com/eliorsulem/HSplit-corpus
[8] https://github.com/cocoxu/simplification/tree/master/data/turkcorpus

For our evaluation, we used a parsing and regeneration procedure: each graph was transformed into sub-graphs. We fed the top parse for each sub-graph as input to the ACE generator, to finally recombine the sentences.

4.2. Evaluations Metrics

For the automatic evaluation of GRASS according to the following state-of-the art metrics we used the EASSE package [45]:

(1) BLEU [48] relies on the proportion of n-gram matches between a system's output and references.

(2) SARI [22] compares the n-grams of the system output with those of the input and the human references, separately evaluating the quality of words that are added, deleted, or kept by a system.

(3) SAMSA [49] measures structural simplicity (i.e., sentence splitting), in contrast to SARI, which is designed to evaluate simplifications involving paraphrasing.

In addition, Quality Estimation Features leverages both the source sentence and the output simplification to provide additional information on simplification systems, in particular: (4) the average number of sentence splits performed by the system, (5) the proportion of exact matches (i.e., original conserved sentences).

5. EXPERIMENT RESULTS

Applying GRASS to the 359 sentences of the TurkCorpus, as others syntactical simplification systems have done, we obtain 91 transformed sentences by our transformation rules. On these 359 sentences, 268 sentences were not changed when applying our rules. First, 265 sentences are not transformed because they are syntactically simple and cannot be simplified any further. Example: *Admission to Tsinghua is extremely competitive*. Finally, three other sentences that are syntactically complex are not transformed due to different reasons: (i) no rule has been applied on one sentence; (ii) a sentence has not been parsed by ACE parser, and (iii) a sentence that has been parsed and transformed but not generated by ACE generator.

As our system cannot transform sentences that do not contain the targeted syntactical constructions, we can consider that our system performs the transformation of 91 out of 94 sentences. We compared the transformed 91 sentences to the same ones obtained by the following systems. The outputs of these systems are collected from EASSE[9] [45].

- Two semantic-based syntactic simplification DSS [11] and HYBRID [10].
- Phrase-based Machine Translation (PBMT-R) [46]. The outputs are collected from DRESS repository[10].
- Sentence Simplification with Deep Reinforcement Learning (DRESS-LS) [47].
- Unsupervised Neural Text Simplification UNTS [25].

Results presented in Table 1 show that for these specific metrics computed by EASSE, GRASS obtains higher BLEU, SARI and SAMSA scores than semantic-based, Phrase-based MT and Neural-based text simplification systems. GRASS gets lower additions and deletions proportions because it doesn't deal with lexical simplification and other rewriting operations.

While recent improvement in text simplification has been achieved by the use of neural MT (NMT) approaches, sentence splitting operation has not been addressed by these systems,

[9] https://github.com/feralvam/easse

[10] https://github.com/XingxingZhang/dress/tree/master/all-system-output/WikiLarge/test

potentially due to the rareness of this operation in the training corpora [22]. Indeed, experimenting with a neural system [47][25], these systems present the higher score of unchanged input sentences (conservatism) and lower score of splitting sentences (0.13 and 0.12 for DRESS-LS and UNTS respectively), comparing to semantic-based systems.

Table 1. Automatic evaluation for text simplification systems for the 91 transformed sentences.

Metrics	**GRASS**	**DSS**	**HYBRID**	**PBMT-R**	**DRESS-LS**	**UNTS**
BLEU	**63.85**	62.49	25.65	60.23	43.06	48.0
SARI	**48.81**	48.03	25.04	36.24	38.10	32.4
SAMSA	**51.44**	48.13	30.86	33.54	25.445	26.69
Sent. splits	2.01	**2.53**	0.98	1.04	0.99	1.01
Exact copies	**0.0**	0.01	0.04	0.08	0.13	0.12

Table 2 and 3 give two examples of these systems outputs of the test corpus. Each system splits the original sentence in a specific manner (e.g., DSS splits "more" but not "better").

DSS and GRASS split the first sentence into 3 fragments. The second sentence is split into 5 fragments by DSS, while GRASS system splits it into 3 fragments. As we can see, the sentences obtained by DSS are not simpler than the original one, they are not semantically correct, and they are agrammatical. GRASS splits sentences into semantically and syntactically correct constructions. HYBRID did not split the sentences; it rewrote them by removing parts making the sentences linguistically incorrect and changing their original meanings. Finally, the translation-based system (PBMT-R) is conservative for the two sentences. Neural-based systems simplify sentence privileging the lexical simplification and deletion operation but not splitting operation.

Table 2. System outputs for example 1 of the test sentences.

EXAMPLE 1	
Original	*The tarantula, the trickster character, spun a black cord and, attaching it to the ball, crawled away fast to the east, pulling on the cord with all his strength.*
Hybrid	The tarantula, the trickster character, a black spun cord, and it attaching, crawled, pulling all.
DSS	the tarantula the trickster character spun a black cord . attaching it to the ball . character crawled away fast to the east . character pulling on the cord with all his strength .
PBMT-R	The Spider, the trickster character, made a black cord and attached to the ball, crawled away fast to the east, pulling on the cord, with all his strength.
DRESS-LS	The tarantula, the trickster character, spun a black cord and, holding it to the ball.
UNTS	The spider, the trick character, spun a black cord,
GRASS	The tarantula is the trickster character. The tarantula spun a black cord. Attaching it to the ball, the tarantula crawled away fast, to the east. The tarantula pulled on the cord, with all of his strength.

Table 3. System outputs for example 2 of the test sentences.

EXAMPLE 2	
Original	*Following the drummers are dancers, who often play the sogo (a tiny drum that makes almost no sound) and tend to have more elaborate — even acrobatic — choreography.*
Hybrid	Dancers, play the sogo (a drum that no and to .
DSS	the drummers are . dancers often play the sogo (a tiny drum makes almost no sound) . drum makes almost no sound) . the sogo tend to . the sogo have more elaborate even acrobatic choreography .
PBMT-R	Following the drummers are dancers, who often play the sogo (a small drum that makes almost no sound) and tend to have more elaborate -- even acrobatic -- choreography.
DRESS-LS	Following the drummers are dancers, who often play the sogo (a small drum that makes almost no sound).
UNTS	Following the musicians are dancers, who often play the Sogo (a tiny drum that makes almost no sound) and tend to have more happy even - .
GRASS	Dancers, which, play the sogo, often, are following the drummers. The sogo is a tiny drum, which, makes almost no sound. The dancers tend to have more elaborate, even acrobatic choreography.

To compare the semantic-based operation and while Hybrid and DSS deal essentially the coordination and relative clauses, we see that passive forms, appositive and subordination clause are not handled. As we can see, GRASS covers a wider range of syntactic structures and that is due to the choice of semantic representation formalism. DMRS is suited for Natural Language Understanding tasks: unlike UCCA, DMRS has a specific label for proper name; so, in generation, proper names are recognized, and the first letter is capitalized. DMRS gives information about verb mode and tense, our rules are defined in a way that they enable to conjugate the verb in the right tense after splitting.

Finally, while DSS does "more" sentence splitting than other systems, that does not mean that it splits them "better". One of the disadvantages of automatic measures like SAMSA or the average number of sentence splits is that they count the number of ending points in an output without considering the syntactic and semantic aspects in the sentence. DSS has high score for SAMSA and for the number of splitting. However, the meaning is not always kept, and the output does not preserve the Subject-Verb-Object (SVO) order. The important number of splitting doesn't mean that the system performs better, yet it is considered as such following the automatic metrics.

6. CONCLUSIONS

In this paper, we have presented GRASS, an automatic syntactic simplification system for English based on semantic representations. To implement our system, we used different available NLP tools performing parsing, graph generation, visualization, and sentence rewriting. After a comparison with stablished state-of-the-art similar methods, our system outperforms particularly on rewriting shared elements on the 359 sentences of TurkCorpus as other existing syntactic simplification systems. Our system also provides a better coverage of syntactic constructions and provides interpretability of the syntactic transformations. We have run an automatic evaluation that shows that GRASS has better scores on BLEU, SARI and SAMSA scores as regards to other existing systems. On this TurkCorpus corpus reduced to 359 sentences we are currently running a human evaluation campaign that will provide a more fine-grained

linguistic analysis of the data obtained with our system. However, the evaluation of our system should be done on a larger corpus than the TurkCorpus limited to 359 sentences, in which only 94 sentences are concerned by the transformations defined in GRASS. We hope to be able to evaluate our system, mainly automatically, on a larger corpus: the complete TurkCorpus, but also other corpora like the Newsela corpus.

In the future we would also like to couple our syntactic simplification system with an existing lexical simplification system based on neural techniques, which would allow us to compare our system with other simplification systems, and to measure the impact of combining these two levels of simplification.

ACKNOWLEDGEMENTS

The authors would like to thank Bruno Guillaume, and Guy Perrier for their support on GREW, and Bastien Gastinel, Hamza Ghorfi and William Domingues for their technical contributions to the development of GRASS.

REFERENCES

[1] Saggion, H. (2017). Automatic text simplification: Synthesis lectures on human language technologies, vol. 10 (1). *California, Morgan & Claypool Publishers*.

[2] Štajner, S., & Popović, M. (2016). Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation* (pp. 230-242).

[3] Niklaus, C., Bermeitinger, B., Handschuh, S., & Freitas, A. (2017). A sentence simplification system for improving relation extraction. *arXiv preprint arXiv:1703.09013*.

[4] Vanderwende, L., Suzuki, H., Brockett, C., & Nenkova, A. (2007). Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, *43*(6), 1606-1618.

[5] Rello, L., Baeza-Yates, R., Bott, S., & Saggion, H. (2013, May). Simplify or help? Text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility* (pp. 1-10).

[6] Sauvan, L., Stolowy, N., Aguilar, C., François, T., Gala, N., Matonti, F., ... & Calabrese, A. (2020, May). Text simplification to help individuals with low vision read more fluently. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)* (pp. 27-32).

[7] Siddharthan, A. (2002, December). An architecture for a text simplification system. In *Language Engineering Conference, 2002. Proceedings* (pp. 64-71). IEEE.

[8] Copestake, A. (2009, March). Invited Talk: Slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)* (pp. 1-9).

[9] Sulem, E., Abend, O., & Rappoport, A. (2018). BLEU is not suitable for the evaluation of text simplification. *arXiv preprint arXiv:1810.05995*.

[10] Narayan, S., & Gardent, C. (2014, June). Hybrid simplification using deep semantics and machine translation. In *The 52nd annual meeting of the association for computational linguistics* (pp. 435-445).

[11] Sulem, E., Abend, O., & Rappoport, A. (2018). Simple and effective text simplification using semantic and neural methods. *arXiv preprint arXiv:1810.05104*.

[12] Chandrasekar, R., Doran, C., & Bangalore, S. (1996). Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

[13] Siddharthan, A. (2002, December). An architecture for a text simplification system. In *Language Engineering Conference, 2002. Proceedings* (pp. 64-71). IEEE.

[14] Brouwers, L., Bernhard, D., Ligozat, A. L., & François, T. (2014, April). Syntactic sentence simplification for French. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)@ EACL 2014* (pp. 47-56).

[15] De Belder, J., & Moens, M. F. (2010). Text simplification for children. In *Prroceedings of the SIGIR workshop on accessible search systems* (pp. 19-26). ACM; New York.

[16] Siddharthan, A. (2014). A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, *165*(2), 259-298.

[17] Shardlow, M. (2014). A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, *4*(1), 58-70.

[18] Shieber SM, Schabes Y. Synchronous tree-adjoining grammars.

[19] Siddharthan, A., & Mandya, A. A. (2014). Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*. Association for Computational Linguistics.

[20] Candido Jr, A., Maziero, E. G., Specia, L., Gasperin, C., Pardo, T., & Aluisio, S. (2009, June). Supporting the adaptation of texts for poor literacy readers: a text simplification editor for brazilian portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 34-42).

[21] Candido Jr, A., Maziero, E. G., Specia, L., Gasperin, C., Pardo, T., & Aluisio, S. (2009, June). Supporting the adaptation of texts for poor literacy readers: a text simplification editor for brazilian portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 34-42).

[22] Xu, W., Napoles, C., Pavlick, E., Chen, Q., & Callison-Burch, C. (2016). Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, *4*, 401-415.

[23] Scarton, C., & Specia, L. (2018, July). Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 712-718).

[24] Zhu, Z., Bernhard, D., & Gurevych, I. (2010, August). A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)* (pp. 1353-1361).

[25] Surya, S., Mishra, A., Laha, A., Jain, P., & Sankaranarayanan, K. (2018). Unsupervised neural text simplification. *arXiv preprint arXiv:1810.07931*.

[26] Narayan, S., & Gardent, C. (2015). Unsupervised sentence simplification using deep semantics. *arXiv preprint arXiv:1507.08452*.

[27] Todirascu, A., Wilkens, R., Rolin, E., François, T., Bernhard, D., & Gala, N. (submitted) HECTOR: A Hybrid TExt SimplifiCation TOol for Raw text in French. Current submission to LREC 2022.

[28] Kamp, H. (2013). A theory of truth and semantic representation. In *Meaning and the Dynamics of Interpretation* (pp. 329-369). Brill.

[29] Narayan, S., Gardent, C., Cohen, S. B., & Shimorina, A. (2017). Split and rephrase. *arXiv preprint arXiv:1707.06971*.

[30] Gardent, C., Shimorina, A., Narayan, S., & Perez-Beltrachini, L. (2017, July). Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

[31] Mendes, P. N., Jakob, M., & Bizer, C. (2012). *DBpedia: A multilingual cross-domain knowledge base* (pp. 1813-1817). European Language Resources Association (ELRA).

[32] Abend, O., & Rappoport, A. (2013, August). Universal conceptual cognitive annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 228-238).

[33] Nisioi, S., Štajner, S., Ponzetto, S. P., & Dinu, L. P. (2017, July). Exploring neural text simplification models. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 85-91).

[34] Flickinger, D. (2000). On building a more effcient grammar by exploiting types. *Natural Language Engineering*, *6*(1), 15-28.

[35] Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal recursion semantics: An introduction. *Research on language and computation*, *3*(2), 281-332.

[36] Emerson, G., & Copestake, A. (2015). Leveraging a semantically annotated corpus to disambiguate prepositional phrase attachment. Association for Computational Linguistics.

[37] Horvat, M. (2017). *Hierarchical statistical semantic translation and realization* (No. UCAM-CL-TR-913). University of Cambridge, Computer Laboratory.

[38] Yao, X., Bouma, G., & Zhang, Y. (2012). Semantics-based question generation and implementation. *Dialogue & Discourse*, *3*(2), 11-42.

[39] Kuhnle, A., & Copestake, A. (2017). Shapeworld-a new test methodology for multimodal language understanding. *arXiv preprint arXiv:1704.04517*.

[40] Kramer, J., & Gordon, C. (2014, August). Improvement of a naive Bayes sentiment classifier using MRS-based features. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (* SEM 2014)* (pp. 22-29).

[41] Guillaume, B., Bonfante, G., Masson, P., Morey, M., & Perrier, G. (2012, June). Grew: un outil de réécriture de graphes pour le TAL (Grew: a Graph Rewriting Tool for NLP)[in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 5: Software Demonstrations* (pp. 1-2).

[42] Bonfante, G., Guillaume, B., & Perrier, G. (2018). *Application of Graph Rewriting to Natural Language Processing*. John Wiley & Sons.

[43] Guillaume, B. (2021, April). Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations* (pp. 168-175).

[44] Goodman, M. W. (2019, October). A Python library for deep linguistic resources. In *2019 Pacific Neighborhood Consortium Annual Conference and Joint Meetings (PNC)* (pp. 1-7). IEEE.

[45] Alva-Manchego, F., Martin, L., Scarton, C., & Specia, L. (2019). EASSE: Easier automatic sentence simplification evaluation. *arXiv preprint arXiv:1908.04567*.

[46] Wubben, S., Van Den Bosch, A., & Krahmer, E. (2012, July). Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1015-1024).

[47] Zhang, X., & Lapata, M. (2017). Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*.

[48] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).

[49] Sulem, E., Abend, O., & Rappoport, A. (2018). Semantic structural evaluation for text simplification. *arXiv preprint arXiv:1810.05022*.

Authors

Rita Hijazi is a PhD student at Aix-Marseille University, France, since 2019, in co-direction between the Laboratoire Parole et Langage (Speech and Language Laboratory), LPL UMR7309 and the Laboratoire Informatique et Systèmes (Computer Science and Systems Laboratory) LIS UMR7020. She has a Bachelor's degree in Linguistics and a Master's degree in Natural Language Processing from the Lebanese University, Lebanon. Her research interests involve NLP tasks like Automatic Text Simplification.

Bernard Espinasse obtained his PhD in 1981 from the University of Aix-Marseille (AMU) after an Engineer diploma from the Ecole Nationale Supérieure des Arts et Métiers of Paris in 1977. He was Assistant Professor at Laval University in Quebec (Canada) from 1983 to 1987. He is currently Full Professor at AMU and researcher at LIS UMR CNRS 7020 lab., where he was team leader for more than fifteen years. He is the author of numerous publications in various fields of computer science, particularly in text mining.

Núria Gala is Assistant Professor at Aix Marseille Univ. (AMU, France) since 2004 and researcher at the Laboratoire Parole et Langage (LPL UMR 7309) since 2017. She is interested in analyzing linguistic complexity and in building resources to help struggling readers improve reading and vocabulary learning. Her research projects are oriented towards the use of language technologies in computer-assisted language learning applications, and towards populations with special reading-comprehension needs (low-readers, dyslexic readers, illiterates, etc.). She is the author of numerous publications in computational linguistics, and natural language processing.