Accepted Manuscript

Please cite this article as: García-Méndez, S., Fernández-Gavilanes, M., Costa-Montenegro, E., Juncal-Martínez, J., & Javier González-Castaño, F. (2019). A library for automatic natural language generation of spanish texts. *Expert Systems with Applications*, *120*, 372-386. doi:10.1016/j.eswa.2018.11.036

Link to published version: https://doi.org/10.1016/j.eswa.2018.11.036

A Library for Automatic Natural Language Generation of Spanish Texts

Silvia García-Méndez[a,*], Milagros Fernández-Gavilanes[a], Enrique Costa-Montenegro[a], Jonathan Juncal-Martínez[a], F. Javier González-Castaño[a]

[a] *GTI Research Group, Telematics Engineering Department, University of Vigo, EI Telecomunicación, Campus, 36310 Vigo, Spain*

Abstract

In this article we present a novel system for *natural language generation* (NLG) of Spanish sentences from a minimum set of meaningful words (such as nouns, verbs and adjectives) which, unlike other state-of-the-art solutions, performs the NLG task in a fully automatic way, exploiting both knowledge-based and statistical approaches. Relying on its linguistic knowledge of vocabulary and grammar, the system is able to generate complete, coherent and correctly spelled sentences from the main word sets presented by the user. The system, which was designed to be integrable, portable and efficient, can be easily adapted to other languages by design and can feasibly be integrated in a wide range of digital devices. During its development we also created a supplementary lexicon for Spanish, *aLexiS*, with wide coverage and high precision, as well as syntactic trees from a freely available definite-clause grammar. The resulting NLG library has been evaluated both automatically and manually (annotation). The system can potentially be used in different application domains such as augmentative communication and automatic generation of administrative reports or news.

Key words: Natural language generation; Spanish; text planning; lexicon; labelled text corpora; Augmentative and Alternative Communication.

1. Introduction

Natural language generation (NLG) has attracted increasing interest in the field of human-computer interaction, as it responds to the demand for coherent and natural-sounding fully machine-generated texts. NLG was for some time considered a sub-field of natural language processing (NLP). However, due to its growing significance and the fact that it requires expertise in various research areas, including linguistics and computation, it has evolved into a major research topic and a discipline in its own right. It has been defined as "*[...]the sub-field of artificial intelligence (AI) and computational linguistics that focuses on computer systems that can produce understandable texts in English and other human languages, typically starting from some non-linguistic representation of information as input[...]*" (Reiter & Dale, 2000). Nevertheless, this definition may be considered obsolete since, as we will explain later, the input to NLG systems consists not only of non-linguistic information like objective data, but also linguistic information (words, sentences, texts) and even visual data.

Traditionally, NLG focused on text-to-text generation, regarding which many sub-fields existed, such as summarizing, text simplification and automatic question generation. The earliest systems took inputs like words, sentences and even whole texts to produce new text as output. However, new data-driven methods have expanded the possibilities of NLG. Most data-to-text generation methods rely on predefined templates to automatically transform data into text by filling gaps in predefined text templates, which has applications in reportage of weather, traffic, sports, health, etc. The more recent vision-to-text systems (Thomason et al., 2014) produce texts, mainly using deep-learning approaches, from visual representations of information such as pictures.

It is broadly agreed that NLG has only recently begun to take full advantage of recent advances in data-driven, machine learning and deep-learning techniques.

NLG tasks are generally addressed by splitting them into sub-problems (Reiter & Dale, 2000, 1997): content determination (deciding which events are important), text structuring (ordering information in the output text), sentence aggregation, lexicalization (finding the right words and sentences to express information), referring expression generation (identification of domain objects) and linguistic realization (generation of well-formed texts). New general systems and applications follow these trends in a broad sense, described in more detail in what follows.

Content determination consists of selecting which information should be included and excluded in the process of text generation. It may be seen as a filtering process and is clearly context/application-dependent. The result is abstracted into semantic, formal, logical and graph

*Corresponding author: sgarcia@gti.uvigo.es
Email addresses: sgarcia@gti.uvigo.es (Silvia García-Méndez), mfgavilanes@gti.uvigo.es (Milagros Fernández-Gavilanes), kike@gti.uvigo.es (Enrique Costa-Montenegro), jonijm@gti.uvigo.es (Jonathan Juncal-Martínez), javier@det.uvigo.es (F. Javier González-Castaño)

structures. Researchers have started exploring data-driven techniques for this sub-problem (Kutlak et al., 2013). Text structuring and discourse planning is the ordering of sentences or paragraphs in how they are presented to readers. The importance of individual events (sentence/paragraph) for the final audience is assessed, taking into consideration internal relations, strongly dependent on the application domain. State-of-the-art solutions include manual rule-based approaches (Mairesse & Walker, 2007; Dušek & Jurcicek, 2015) and rhetorical structure theory (RST) (Williams & Reiter, 2008), machine learning techniques (Lampouras & Vlachos, 2016; Mei et al., 2015), systemic functional grammar (SFG) (Bateman, 1997), meaning-text theory (MTT) (Wanner et al., 2010) and the centred theory/approach (Barzilay & Lapata, 2008), among others. Sentence aggregation at the semantic or syntactical level, which tries to join data into single sentences, deals with fluency and readability; however, conceptually the approach is complex. Lexicalization converts the result of the previous stage into natural language (NL) but has the problem that there may exist many ways to express the same idea in NL. Logically, the more possibilities the system can choose from, the better. While a simple approach is to convert domain concepts into lexical items, sometimes the task is made more complex due to gradable properties, such as size and colour. Referring expression generation (REG) consists of selecting the words or phrases that can describe domain entities in an unambiguous manner. Selected is the best set of known properties to distinguish an element from others, while any information that is not directly relevant to the identification task is discarded. Several algorithms can be found in the literature for this purpose including the full brevity procedure (Dale, 1989), the greedy heuristic algorithm (Dale, 1992; Frank et al., 2009) and the incremental algorithm (Dale & Reiter, 1995). Finally, linguistic realization involves text ordering, the generation of morphological forms, the insertion of function words like auxiliary verbs and prepositions and the insertion of punctuation marks. Here the *generation gap* appears, because sometimes it is necessary to add elements that are not present in the input data. Templates are widely used to address this issue but only when the domain is small and the expected variation is minimal. These approaches yield better quality than other approaches but are very time consuming and do not scale well enough in certain applications with high linguistic variation. Hand-coded grammar-based systems are the alternative to templates but they require very detailed input such as KPML (Bateman, 1997) based on systemic-functional grammar (SFG). Other alternatives include statistical approaches, which derive probabilistic grammar from large corpora, reducing effort while increasing coverage (Langkilde & Knight, 2002); the head-driven phrase structure grammar (HPSG) (Nakanishi et al., 2005), the lexical-functional grammar (LFG) (Cahill & Van Genabith, 2006) and the tree-adjoining grammar (TAG) (Gardent & Narayan, 2015). We follow a hybrid approach that exploits the advantages of grammar-based and stochastic systems but also reduces the effort of the NLG process.

As noted in Stent et al. (2005), generated quality depends on adequacy, fluency, readability and variation. We can distinguish three main NLG architectures: traditional modular architectures (macro-planning or selection and text structuring, micro-planning or sentence aggregation, lexicalization, referring expression generation, and linguistic realization applying syntactic and morphological rules), planning perspective architectures (less modular but also with roots in the AI tradition) and data-driven approaches. The latter rely on statistical learning and represent a strong trend in NLG, but the most widely adopted approach today is the rule-based (or template-based) method (Cheyer & Guzzoni, 2007; Mirkovic & Cavedon, 2008).

The trade-off between output quality and efficiency is becoming a central issue. Recent years have witnessed a marked interest in automatic text generation, where "automatic" means that the user is only required to introduce meaningful words like nouns, verbs and adjectives.

We are interested in automatically generating complete, coherent and correctly spelled sentences from specific content specified by the user at word level as "key points" (verbs, nouns and adjectives). Our practical approach to text-to-text generation is easily adaptable to other languages and integrable in a wide range of digital devices. This work is not our first attempt at developing an automatic NLG system for Spanish. In García-Méndez et al. (2018) we described an automatic version of SimpleNLG for Spanish but soon realized that it was difficult to expand and improve. This new automatic NLG system is based on a modular architecture that allows domain-dependent components to be separated from domain independent components.

The rest of this paper is organized as follows. First we review the state-of-the-art for NLG (Section 2). We then describe a hybrid system combining linguistic rule-based and statistical techniques, composed of two subsystems: a knowledge base consisting of the *aLexiS* Spanish lexicon (Section 3.1) and a grammar (Section 3.2). We then describe an interface engine (Section 3.3), followed by our evaluation results (Section 4). To test the library we created a dataset specially tailored for automatic NLG (Section 4.2) and used manual and automatic procedures (Section 4.3). We compared our system with an automatic Spanish version of the SimpleNLG library (Section 4.5). Finally, in Section 5 we conclude the paper.

2. Related work

NLG started in the second half of the 20th century with automatic translation (Sager, 1967). Research in the 1970s focused on choosing appropriate words to express abstract conceptual content and using these to generate appropriate textual structures (Schank, 1975; Mann, 1982). It was not until the 1980s when NLG was truly recognized beyond language understanding in reverse mode. A number of significant developments occurred during this decade (McKeown, 1985; Appelt, 1985), marked by the move away from large monolithic systems that attempted to resolve all NLG stages of specific problems. However, by the end of the 1980s substantial research adopted an AI perspective. During the 1990s there were significant new achievements (Reiter & Dale, 1997; Hovy, 1993) and the first real-world NLG applications appeared, including the pioneering FoG weather forecasting system (Goldberg et al., 1994). In addition, the interest in multilingual generation grew.

Nowadays, NLG systems are considered to be highly sensitive to problem characterization and, in most cases, are purpose-built. Conversely, our system is based on a modular architecture that allows us to separate domain-dependent (grammar and lexica) from domain-independent components (NLG surface realizer). This means that it can be adapted to different purposes and fields of interest using specific syntactic structures and vocabulary.

From a practical perspective, NLG is capable of generating rich and coherent texts, nearly indistinguishable from those created by humans, that is, satisfying completeness (containing enough meaningful information), grammatical and orthographic correctness and semantic coherence criteria.

If we consider generated text types, we can mainly distinguish between informative texts, summaries, simplified texts, persuasive texts, dialogues, recommendations and creative texts.

The systems to generate informative texts from objective data, e.g. SumTime (Reiter et al., 2005), which does not perform text-to-text generation and is only available for English, are highly language-sensitive and so are only suitable for the language they were designed for. Most of these systems generate routine documents (Reiter et al., 1995), such as administrative documents, response letters, operating manuals, weather forecasts (Belz, 2008) and traffic and academic reports. At the other extreme, more creative texts are more challenging to generate. Experiments in this line based on predefined templates are very rigid (Peinado et al., 2004). We seek to develop a tool that should not constrain the content of the generated sentences but assist users to express themselves with the words that come to mind without loss of application generality. Affective NLG systems can, in fact, generate texts beyond factual information to create, among other possibilities, poetry (Gervás, 2001), but this goes beyond our purposes and, with these systems, users have no control over content generation.

Summary generation has applications in medicine, finance and sport. Persuasive texts try to influence user behaviours; a representative example is STOP (Reiter et al., 2003) to discourage smoking. Dialogue systems focus on human-machine communication and are of interest for call centres or games (Koller et al., 2010). Some aim at generating explanations as sequences of steps (for instance, P.REX (Fiedler, 2005), a tool to generate demonstrations of algorithms). Most of these proposals, like the Shed (Lim-Cheng et al., 2014) recommender for personalized nutritional plans and the text-to-text generator of Wikipedia articles from Internet documents in Sauper & Barzilay (2009), require an infeasible amount of time and resource-consuming predefined templates to address generality, which also leads to lack of control over content generation. Flights (White et al., 2010) is an example of a hybrid tool, which uses templates to organize the input data and the OpenCCG[1] framework to generate the final user-oriented text, relying on n-gram models and language factorized grammars (FLM). Nevertheless, one of its drawbacks is that it is too specific due to its training process and complex specification of input data. Summing up, templates are still the basis for the main NLG approaches, except for some recent systems that apply statistical techniques such as deep learning and that are highly domain-dependent.

We conclude this review by mentioning the most relevant systems nowadays. The SimpleNLG library (Gatt & Reiter, 2009) conducts a surface realization task using a knowledge-based approach. It is available for English, with versions also for Brazilian Portuguese (De Oliveira & Sripada, 2014), French (Vaudry & Lapalme, 2013), German (Bollmann, 2011) and Italian (Mazzei et al., 2016). This library has influenced the NLG field strongly due to its simple input requirements compared to other systems. Its main drawback is that it is not automatic. NaturalOWL (Androutsopoulos et al., 2014) is a data-to-text tool that generates texts from an ontology (OWL) (it does not perform text-to-text generation, just data-to-text generation). In this line, in Dale et al. (2005) the authors presents a tool to generate route descriptions. Unlike SimpleNLG, the input format happens to be very complex, defined by a "route planning markup language". The method in Liu et al. (2017) to create radiology reports automatically relies on templates. Conversely, the system in Chen et al. (2002) is composed of a trainable sentence planner and a stochastic surface realizer, similar to our approach, but only available for English.

To the best of our knowledge no other system performs text-to-text generation in an automatic way, regardless of the target language (Spanish or other). Shed and the system by Gervás (2001) do not allow the user to control content generation. Others like OpenCCG cannot be easily used for other than the purpose for which they were designed or have a complex interface for data input.

In García-Méndez et al. (2018) we described our automatic Spanish version of SimpleNLG, but, since it was difficult to improve, we created an automatic NLG system based on a modular architecture. Furthermore, without loss of generality, we identified a new use case for NLG to help people diagnosed with communication disorders such as autism spectrum disorder (ASD) to express themselves more easily and quickly. The pictograms in their personal communicators may represent concepts or single words, and the users determine the key words that serve as input

[1] Available at `http://openccg.sourceforge.net/`.

to create a coherent and complete sentence. In this use case a template-oriented approach would be unacceptably limiting.

3. Methodology and system architecture

Below we explain our methodology and the resulting system architecture. Firstly, we focus on the morphological part, i.e. the lexicon that provides the system with the indispensable linguistic knowledge (Section 3.1). We then analyse the syntactic stage, which uses a define-clause grammar (DCG) for syntactic structuring (Section 3.2), and we conclude with an overview of the system (Section 3.3).

3.1. The aLexiS *lexicon*

Since we want our NLG system to be fully automatic, not only in selecting the appropriate grammar structure for the words given by the user but also in the inflection of these words, we need a wide vocabulary and its corresponding linguistic data. This is why we created the *aLexiS* lexicon, achieved by interpreting input resources and merging lexica following two well-defined steps (Crouch & King, 2005) with an intermediate verification step. The merging process is automatic, that is, without human supervision, and proceeds as follows:

1. Extraction and mapping: all possible entries are extracted and mapped to a common format.
2. Verification: all extracted and mapped entries are verified at a lexical level to check if they are commonly accepted in the *Diccionario de la Real Academia de la Lengua Española*[2] (DRAE), the reference dictionary for the Spanish language. Their lexical categories are also checked.
3. Combination: the new resource is created from lexica that are compared and combined automatically.

3.1.1. Linguistic resources

We used existing Spanish linguistic resources to build *aLexiS*. This seemed a good start to developing a new lexicon, which required interpreting the different input data. We looked for freely available resources in terms of access, modification and distribution. We also gave importance to correctness of the entries and wide coverage. These existing linguistic resources were:

- *Lexicon of Spanish Inflected Forms*[3] (LEFFE), a morphological and syntactic lexicon with wide coverage (Molinero et al., 2009) based on the Alexina framework (Sagot, 2010). It follows the linguistic criteria applied in the equivalent lexicon for French LEFFF, taking advantage of the linguistic proximity between Spanish and French as Romance languages (Sagot, 2010).

[2]Available at `http://www.rae.es/`.

[3]Available at `https://gforge.inria.fr/frs/?group_id=482&release_id=4290`, Nov. 2016.

```
aposento 100 n
    [pred="aposento_1, cat=n,
    <Objde:(de-sn|de-sinf),
    Obja:(a-sinf)>", @CMS000]
    aposento_1 Default CMS000 %default
aposentos 100 n
    [pred="aposento_1, cat=n,
    <Objde:(de-sn|de-sinf),
    Obja:(a-sinf)>", @CMP000]
    aposento_1 Default CMP000 %default
```

(a) In LEFFE.

```
aposento - nombre masculino
    singular aposento
    plural aposentos
    forma nominal de:  aposentar
    Palabra atestada en DRAE
```

(b) In *OSLIN-es*.

Figure 1: Example of the Spanish lemma *aposento* 'bedroom' in both resources.

- *OSLIN-es*[4], a large-scale lexicon that includes words attested by DRAE (Janssen, 2005, 2009).

3.1.2. Extraction and mapping

Extraction and mapping were performed separately; first, information was extracted from different resources and then it was mapped to a common format. We extracted some entries from LEFFE to start with. Each extracted LEFFE word entry was represented in the extensional Alexina format (Danlos & Sagot, 2008). For example, Figure 1.a represents the LEFFE word entry for the Spanish lemma *aposento* 'bedroom'. We chose verbs and nouns, because they play the most important role in sentences. We also extracted entries tagged as adjective, adverb, determinant, pronoun, conjunction and preposition, but discarded those tagged as interjection, numeral and proper name.

In the example it can be observed that the lemma (after the word `pred`) is a noun (represented in LEFFE with `C` and `cat=n`) with two possible forms: masculine singular (`CMS000`) for *aposento* 'bedroom' and plural (`CMP000`) for *aposentos* 'bedrooms'.

After extraction, in order to obtain more related entries we obtained the associated lemma for each inflectional form in LEFFE. Each lemma was searched automatically using the online OSLIN-es morphological database (Janssen, 2005). Both the forms and their morphological information and other related lemmas were retrieved. For example, Figure 1.b shows the result of the search of lemma *aposento* 'bedroom'. In this case, the retrieved word entry indicates that the lemma is a masculine noun (*nombre masculino*), with two forms: singular and plural. Both forms are also included in LEFFE, but in this case two extra items of information were obtained: the word *aposento* 'bedroom' as the nominal form of *aposentar* 'to lodge', which is not a word entry in LEFFE; and the searched

[4]Available at `http://es.oslin.org/`, Nov. 2016.

lemma with a verified entry in the DRAE. In other words, only some OSLIN-es word entries were verified. For each new word entry detected in OSLIN-es by this procedure, the process was repeated iteratively. In our example, the new search was carried out from the word *aposentar* ‘to lodge’, yielding all the forms for the following Spanish conjugation:

- *Indicativo* ‘indicative’: *presente* ‘present’, *pretérito imperfecto* ‘past simple’, *futuro* ‘future’, *pretérito perfecto* ‘past simple’, *condicional* ‘conditional’;
- *Subjuntivo* ‘subjunctive’: *presente* ‘present’, *pretérito perfecto* ‘past simple’ and *futuro* ‘future’;
- *Imperativo* ‘imperative’;
- Non-finite forms: *infinitivo* ‘infinitive’, *gerundio* ‘gerund’ or ‘present participle’ and *participio* ‘past participle’.

Since Spanish is a highly inflected language with about 50 conjugated forms per verb (Molinero et al., 2009), it was important to retrieve and save the complete conjugation of each verb, omitting the compound verbal forms, because these are built using the conjugation of the auxiliary verb *haber* ‘to have’. This information allowed the verbal tense to be adjusted to the semantic features of a specific context.

The problem with the extraction process is that the word entries have different formats and tags. In order to simplify the merging step and avoid possible future inconsistencies, it was necessary to convert all entries to a common format. Algorithm 1 describes this process.

Algorithm 1 : Extraction and mapping algorithm

```
function EXTRACTION_MAPPING({LEFFE})
    for e_LEFFE ∈ {LEFFE} do
        lem_e_LEFFE = e_LEFFE.getLemma()
        cat_e_LEFFE = e_LEFFE.getCat()
        if lem_e_LEFFE.isInOslin() then
            {e_OSLIN} = searchInOslin(lem_e_LEFFE)
            %Entries in Oslin with lemma morphological data.
            {OSLIN}.add({e_OSLIN})
        end if
    end for
    %Completion of syntactic and semantic data for new entries of
OSLIN not present in LEFFE.
    for e_OSLIN ∈ {OSLIN} do
        lem_e_OSLIN = e_OSLIN.getLemma()
        cat_e_OSLIN = e_OSLIN.getCat()
    end for
end function
```

Algorithm 2 : Verification algorithm

```
function VERIFICATION({SET})
    for e_SET ∈ {SET} do
        lem_e_SET = e_SET.getLemma()
        cat_e_SET = e_SET.getCat()
        if !lem_e_SET.isInDRAE() OR !lem_e_SET.catInDRAE (cat_e_SET)
         then
            {SET}.delete(e_SET)
        end if
    end for
end function
```

Algorithm 3 : Lexicon building algorithm

```
{OSLIN} = {∅}
{LEFFE}= LoadLeffe()
EXTRACTION_MAPPING({LEFFE})
{SETS} = {{LEFFE},{OSLIN}}
for {set} ∈ {SETS} do
    VERIFICATION({set})
end for
{aLexiS} = {LEFFE} ∪ {OSLIN}
```

3.1.3. Verification

This step checks that the quality of word entries is satisfactory. Our solution is an automatic process that checks for the existence of each lemma and its lexical categories. In order to determine if a lemma exists and whether its lexical categories are correct, we searched for the word in DRAE. We chose this for its high coverage and the fact that no training was performed, which allowed us to discard incorrect word entries and increase the precision of our lexicon. Algorithm 2 describes the process.

3.1.4. Merging

A combination step merges the collected entries using the graph unification in Necsulescu et al. (2011) and Bel et al. (2011). This operation is based on set unions of compatible feature values. It allows common information to be validated by adding differential data and excluding inconsistent data. It proceeds as follows:

1. For each common lemma, i.e. a lemma that appears in all lexica, it puts together all lexical entries with the same lemma (homography is taken into account only when lexical categories differ).
   (a) For each entry obtained in (1), a unification process is applied checking all the feature structures included in the entries.
   (b) Once 1.a is done, a new entry is created in *aLexiS*, including the set of feature structures, which contains the common information as well as any particular data in any entry from the different resources.
2. When a lexical entry cannot be joined with any input from the other lexica, a new entry is created in *aLexiS* containing that unique information. The same occurs with lemmas that only belong to one lexicon.

Note that the merging procedure avoids any possible inconsistencies thanks to the common format in the extraction and mapping step. Algorithm 3 shows the sequence of steps in this section.

Therefore, *aLexiS* was built by interpreting inputs extracted from previous resources, verifying them and finally merging them into a common format. Figure 2 shows the result for *aposento* ‘bedroom’. Since the formats of the resources were combined into this final result, the lexical information associated with the lemma and PoS tag remain the same. In the case of word forms, the information is repeated in both LEFFE and OSLIN-es.

```
<Entry lemma='aposento'>
  <feat att='POS' val='n'/>
  <WordForm>
    <feat att='wForm' val='aposento'/>
    <feat att='gender' val='m'/>
    <feat att='number' val='sg'/>
  </WordForm>
  <WordForm>
    <feat att='wForm' val='aposentos'/>
    <feat att='gender' val='m'/>
    <feat att='number' val='pl'/>
  </WordForm>
</Entry>
```

Figure 2: Example of Spanish lemma *aposento* 'bedroom' in *aLexiS* entry.

3.1.5. Automatic lexicon extension

In order to simplify NLG by avoiding the introduction of prepositions as input, we need to infer *a priori* which specific preposition follows a verb. Training was based on a dataset of Spanish novels and nearly 500 fairy tales (Andersen, 2016; Anonymous, 2016; Grimm, 2016), which had been previously POS-tagged with Freeling Tagger[5]. We developed a language model from the training process that considered bigrams and trigrams around verbs using syntactic and semantic knowledge.

Many NLG libraries, including SimpleNLG, use collections of words without duplicates. In our case, considering the size of *aLexiS* and its associated linguistic data, we used an index. Moreover, by doing so our system is able to conduct the whole NLG process more quickly.

3.2. Syntactic structure using a grammar

One of the challenges of computational linguistics is syntactic structuring or parsing, which consists of creating the tree parsing structure from a given sentence. We used a grammar the other way around, in order to infer the syntactic structure of the final desired sentence, for which purpose we used a DCG (Maggiori, 2013). In Gavilanes (2012), a DCG is defined by $G = (N, \Sigma, P, S)$, where the formation rules are:

- $\alpha A \gamma \rightarrow \alpha \omega \gamma$ with $A \in (N \cup \{S\}), \alpha, \gamma \in (N \cup \Sigma)^*, \omega \in (N \cup \Sigma)^* - \{\epsilon\}$

  or

- $S \rightarrow \epsilon$

with $|\alpha A \gamma| \preceq |\alpha \omega \gamma|$, where $|\alpha A \gamma|$ represents the number of symbols in $\alpha A \gamma$. The languages generated with this type of grammar are called context-sensitive languages (CSL). Given all possible tree structures within a grammar, the system selects the most appropriate structure for the input words. We chose a simple Spanish grammar allowing a wide range of basic sentences with low computational effort[6]. This has the advantage that it tries to reduce grammar rules by annotating number and person considerations. Generative grammars (Ruwet et al., 1974) are context-free – for example: `sentence-->nominal syntagm, verbal syntagm` – but, in addition to cases *Yo tengo frío* 'I am cold' and *nosotros tenemos frío* 'we are cold', they may generate the case *yo tenemos frío* 'I are cold'[7]. In order to avoid incorrect sentences like this it is necessary to multiply the number of rules. Instead, however, the grammar of our choice is annotated with number and person features, thereby ensuring that the verb inflection is correct[8]. The final result is:

```
sentence-->nominal syntagm(person, number),
        verbal syntagm(person, number).
```

In the next subsections the notation is as follows. Elements in upper case letters correspond to tree structures and elements in lower case letters represent word constituents, not variables. In the examples a dashed line represents non-direct substitutions as in the case of SN within SN_COORD. Spanish examples of the tree structures are given in italics. Figure 3 shows a complete example of some linguistic rules extracted from the grammar.

3.2.1. Nominal and coordinated nominal syntagm rules

Nominal syntagms are composed of nouns, pronouns and proper names. Nouns may be preceded by a determiner or followed by an adjectival/adverbial and/or prepositional syntagm. A sentence composed of two nominal syntagms with a conjunction in between is called a coordinated syntagm.

3.2.2. Adjectival, adverbial and prepositional syntagm rules

In this case, a noun may be followed or not by an adjectival/adverbial syntagm. These syntagms are also composed of an adverb or an adjective that may be followed by another adjective or adverb, respectively. As in the case of adjectival/adverbial syntagms, a prepositional syntagm may be empty or may be composed of a preposition followed by a nominal syntagm.

3.2.3. Predicate rule

The predicate of a sentence is composed of a verb that may be followed or not by a nominal or coordinated nominal syntagm and an adjectival/adverbial and/or prepositional syntagm. It may also be composed of a verb followed by another verb, such as in the sentence *Yo intento estudiar ciencias* 'I try to study science'; followed by a nominal

[5] A library that provides multiple language analysis services, including probabilistic prediction of categories of unknown words (Atserias et al., 2006; Padró & Stanilovsky, 2012).

[6] `http://PrologDCG-es.sourceforge.net/`.

[7] This sentence is incorrect grammatically speaking due to the wrong inflection of the verb given the subject.

[8] This is known as an augmented grammar (Grishman & Sandoval, 1991).

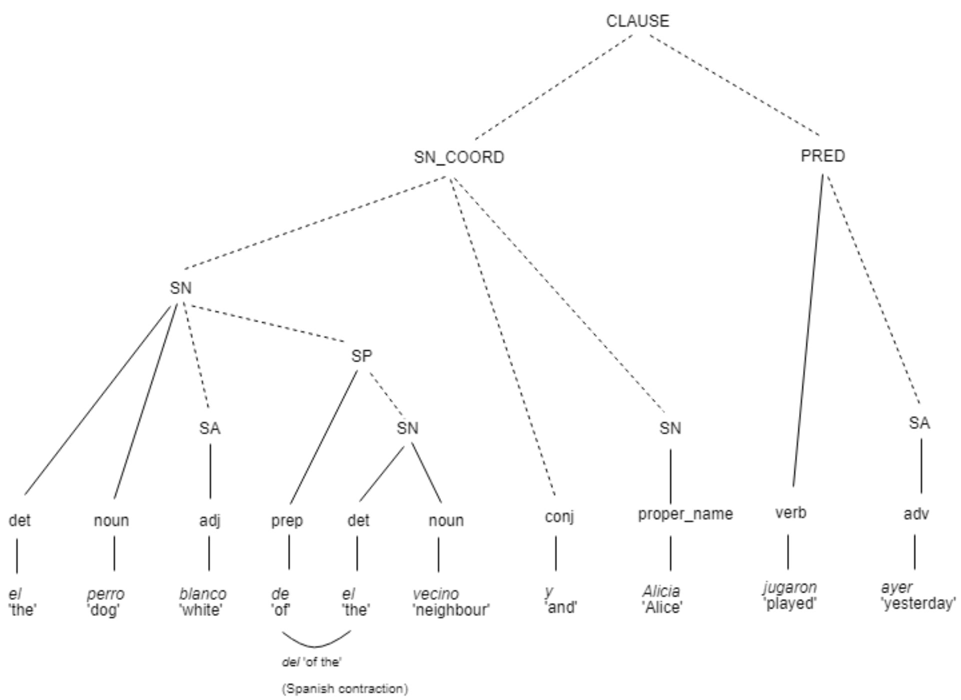

Figure 3: Syntax tree example from the grammar.

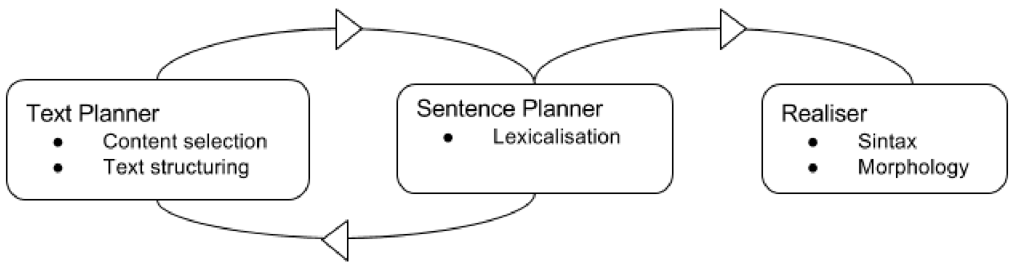

Figure 4: Our three-stage NLG architecture.

or coordinated nominal syntagm and an adjectival/adverbial and/or prepositional syntagm.

3.2.4. Sentence rule

A sentence is composed of a nominal coordinated syntagm followed by a predicate or a single predicate without a subject, which is very common in Spanish. The subject of the sentence may also be a nominal syntagm. Given the relations among the different syntagms, and due to computational and time limitations, we set a depth level of two iterations. Take the nominal syntagm in Figure 3 as an example. The first nominal syntagm includes a prepositional syntagm composed of a nominal syntagm. It is easy to detect the loop condition here. In this regard, the second nominal syntagm cannot be composed of another prepositional syntagm.

3.3. Our NLG library: surface realizer

Our aim is to design an NLG system to generate Spanish sentences from a set of input words. These input words should be meaningful, i.e. nouns, verbs and adjectives. Our system can infer determiners, prepositions and conjunctions automatically to be included in the final sentence.

Our three-stage NLG architecture in Figure 4 is inspired in a state-of-the-art processing pipeline, adapted for automatic Spanish NLG. The first *Text Planner* stage deals with content selection and text structuring. The user selects the words to be considered when creating an NL sentence in Spanish, in SVO (subject-verb-object) order, which is adequate in Spanish for practical NLG applications. The system infers the text structure from the information provided by the grammar. The second *Sentence Planner* stage deals with lexicalization, which comprises the actions of adding words and setting phrases or word patterns. The final *Realizer* stage adds any extra elements needed and performs the morphology inflections to create a coherent and grammatically correct sentence in Spanish.

The main NLG actions are the following. In each case we indicate the stage they belong to:

- Detecting whether a sentence is affirmative, negative or interrogative (*Sentence Planner*). This decision affects the linguistic structure of the sentence. It is taken as negative if it contains the negation adverb *no* 'not'. It is considered interrogative if it contains a question mark (*?*). If the sentence contains both elements the system generates a question in negated form. In any other case the sentence is considered to be affirmative.

- Inclusion of subject (if necessary) (*Sentence Planner*). Sentences with elided subjects are common in Spanish, for example *Voy al parque* 'I go to the park' and *¿Vais al parque?* 'Do you go to the park?', because the inflection of the verbs allows the person and number features of the subject to be identified whether it is elided or not. Bearing in mind that we want the NLG process to be as transparent as possible to the user, we need to minimize the number of words they must provide. When the user does not include any subject, the system takes the first singular personal pronoun *yo* 'I' as the subject of the sentence.

- Syntax structure inference (*Text Planner*). The subject and predicate separation simplifies the search for the syntactic trees that match the input words given by the user. In this way the task is less time/resource-consuming because the trees are smaller. Thus, once the type of sentence (affirmative, negative or interrogative) is established by the *Sentence Planner*, the system separates the subject from the predicate according to the position of the main verb within the sentence and then looks for the best syntactic structure that fits them.

  We conduct a depth-first search (DFS) (Tarjan, 1971) for the best syntactic structures in our grammar given some input words. The search starts at the root and explores each branch as far as possible before backtracking. The DFS algorithm traverses the cumulative syntactic tree in a depth-ward motion and uses a stack to remember the next vertex to start a new search when a dead end is found in any iteration. In the example in Figure 5 DFS traverses from the root retrieving paths 1-2-5, 1-2-6, 1-3, 1-4-7. It employs the following rules.

  - Rule 1. Visit the adjacent unvisited vertex marking it as visited. Then display it. Finally, push it into the stack.
  - Rule 2. If no adjacent vertex is found, pop a vertex up from the stack. All the vertices without adjacent vertices are taken out the stack.
  - Rule 3. Repeat Rule 1 and Rule 2 until the stack is empty.

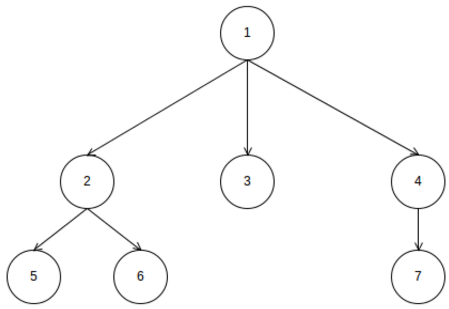

Figure 5: DFS example.

- Addition of any extra elements needed (*Sentence Planner*). This is related to the previous task. Once the syntactic structure has been inferred, it may be necessary to include extra elements such as determiners, prepositions and conjunctions. These elements are included in the sentence if they correspond to feasible realizations in the grammar. This is the reason for the feedback between the first and second stages in Figure 4.

- Morphological inflections (*Realizer*). This encompasses the inflections that are necessary to produce a sentence that is grammatically correct, in which the verb and other constituents are inflected in the way dictated by the subject. These morphological inflections not only deal with conjugation, person, number and gender features, but also with contractions and the Spanish double negation[9]. For example, the negation of the sentence *Yo voy siempre al teatro* 'I always go to the theatre' is *Yo no voy nunca al teatro* 'I never go to the theatre'. The negative meaning clearly results from the adverb *no* 'not' and it implies changing the time adverb from *siempre* 'always' to *nunca* 'never'.

  Since our system distinguishes between the subject and the predicate of a sentence before generating it, it can create sentences with coordinate subject applying appropriate linguistic features to each to adjust number, gender and person features. For example, considering the words *cuidadora, nosotros,*

[9] With a second negation adverb, apart from *no* 'not', reinforcing the negativity of the sentence. This is a notable difference with other languages like English, where a second adverb of negation makes the sentence affirmative.

comer, manzanas ‘caregiver, we, eat, apples’, the resulting sentence *La cuidadora y nosotros comemos manzanas* ‘The caregiver and we eat apples’ has a compound subject.

First, gender, number and person features must be inferred for the whole sentence deriving them from the words given by the user. These features are determined by the subject, which is expected to be a nominal syntagm (coordinated or not). Continuing with the example *La cuidadora y nosotros comemos manzanas* ‘The caregiver and we eat apples’, the subject is a coordinate nominal syntagm composed of two nominal syntagms. The first is composed of a feminine singular determiner *la* ‘the’ and a feminine singular noun *cuidadora* ‘caregiver’. The second nominal syntagm is a pronoun *nosotros* ‘we’. Consequently, the sentence is in third person and plural. This is why the verb *comer* ‘eat’ is inflected that way.

We implemented the number, gender and person linguistic rules for Spanish in DRAE. In principle, the first time the user introduces the words to generate a sentence in NL, the system takes the sentence as masculine, singular and first person, and then, using the information in *aLexiS*, adjusts these features applying the linguistic grammar rules. For example, if the subject is a coordinate nominal syntagm, the sentence is considered to be plural. For the gender, only if all the subject constituents are feminine the sentence is considered to be feminine. Regarding the person, it is necessary to follow the following rules in strict order to make the adjustment:

- 1. If the subject contains an element referring to the first person, the sentence is in the first person.
- 2. If the sentence contains an element referring to the second person that is not related to the first person, the sentence is in the second person.
- 3. If the sentence contains an element referring to the third person that is related to neither the first nor the second person, the sentence remains in the third person.

aLexiS contains the inflections of each lemma according to number and gender changes and, in the case of pronouns and verbs, also according to person features. Once these features are inferred, it is only necessary to apply them to all word inputs. Nevertheless, sometimes there is no subject included and default features should be considered (in our case, as previously mentioned, first person, masculine gender and singular number).

The verbal tense in the final sentence is present unless the user provides a time adverb. For example, if this adverb is *ayer* ‘yesterday’, the sentence is in the past tense. All this linguistic knowledge is taken from *aLexiS*.

Our system also deals with spelling changes due to contractions, usually composed of a preposition and an article or a pronoun. For example, given the words *él, comer, con, yo* ‘he, eat, with, I’, a contraction of the preposition *con* ‘with’ with the pronoun *yo* ‘I’ generates the word *conmigo* ‘with me’. So, the resulting sentence is *Él come conmigo* ‘He eats with me’. The most common contractions in Spanish are *a* ‘to’ + *el* ‘the’ → *al* and *de* ‘of’ + *el* ‘the’ → *del*.

A fully automatic Spanish NLG library requires default rules for atypical situations, such as input words that are not included in the lexicon. In this case they are treated as proper names. The same occurs when no related features are provided or the features cannot be inflected from the input words.

- Orthographic rules (*Realizer*). We implemented the orthographic rules in DRAE.

In order to generate a sentence, it is first necessary to create the syntagms that compose it and then join them paying attention to their syntactical and semantic function. For example, to generate the sentence *La niña juega con el gato* ‘The girl plays with the cat’, first we have to create the nominal syntagm *el gato* ‘the cat’ and integrate it in a prepositional syntagm *con el gato* ‘with the cat’. It is also necessary to build the subject of the sentence *la niña* ‘the girl’ and the predicate with the main verb *jugar* ‘play’. Finally, the subject and the predicate have to be integrated with the prepositional complement in the final sentence. We manage all these stages automatically.

Figure 6 shows another complete example of NLG using our three-stage architecture. The target sentence is *El lobo come niñas* ‘The wolf eats girls’. The user provides the system with the input words: *lobo, comer, niñas* ‘wolf, eat, girls’; which are the meaningful elements in the final NL sentence. In the first stage, the *Text Planner* infers three suitable linguistic realizations for the input. In the second stage, the *Sentence Planner* learns that the sentence is in affirmative mode and adds the extra elements according to the previously selected linguistic realizations in the grammar and the information within *aLexiS*. Finally, in the third stage the *Realizer* conducts the morphological and orthographic processes to generate one or more sentences as result.

Our system inserts conjunctions, determiners and prepositions automatically. In addition, if there is a time-related adverb like *mañana* ‘tomorrow’ among the word inputs, the verbal tense of the sentence is adjusted automatically (in the example, to future tense). In the special case of verbs that can be reflexive or non-reflexive, the system generates the sentence depending on the corresponding probabilities. The system gets this information from our language model in *aLexiS*. We developed an algorithm to build the sentence word by word based on the linguistic

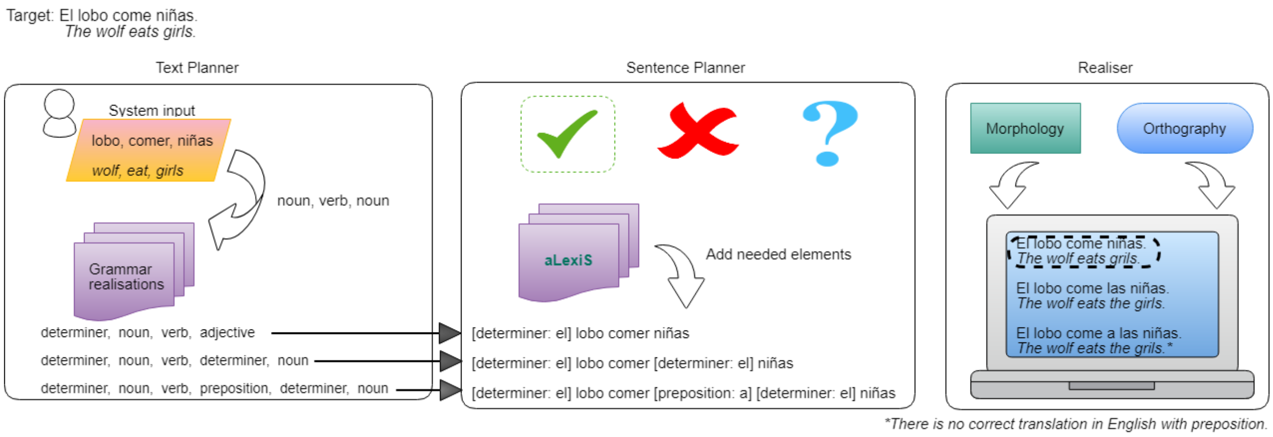

Figure 6: Example of sentence generation using our three-stage NLG architecture.

Input words	Best generated sentence/s
dibujar, animales 'draw, animals'	*Yo dibujo animales.* 'I draw animals.'
Ana, ir, colegio, no 'Ana, go, school, not'	*Ana no va al colegio.* 'Ana doesn't go to school.'
pájaros, poder, volar, ? 'birds, can, fly, ?'	*¿Los pájaros pueden volar?* 'Can birds fly?'
niñas, tomar, batido, chocolate 'girls, have, milkshake, chocolate'	*Las niñas toman el batido del chocolate.* *Las niñas toman el batido y el chocolate.* 'The girls have the chocolate milkshake.' 'The girls have the milkshake and the chocolate.'
profesor, escribir, letras, números, en, pizarra 'teacher, write, letters, numbers, on, blackboard'	*El profesor escribe las letras y los números en la pizarra.* 'The teacher writes the letters and the numbers on the blackboard'.
abejas, volar, alrededor, de, flor, amarillo 'bees, fly, around, flower, yellow'	*Las abejas vuelan alrededor de la flor amarilla.* 'The bees fly around the yellow flower.'

Table 1: Examples of sentences illustrating the functionalities of our NLG library.

knowledge in the lexicon and the grammar rules that we extracted for Spanish.

3.3.1. Functionalities

Our library allows coherent and complete sentences in Spanish to be constructed that can be affirmative, negative or interrogative. It is possible to create simple sentences and complex sentences with compound subject or double negation. Table 1 shows sentences that were created automatically in increasing order of linguistic complexity, as well as the corresponding input words.

4. Quantitative analysis and experimental results

4.1. Lexicon

- *GilcUB-M Dictionary.* A full form dictionary with morphosyntactic annotations in Multext encoding schema conforming with EAGLES standards for morphosyntactic encoding of computational lexica. It contains 62,244 unique (lemma, PoS) pairs[10] (29.12% fewer than *aLexiS*).

- *Freeling.* A morphosyntactic lexicon used for morphological analysis and PoS disambiguation modules in the *FreeLing* NLP tool (Padró et al., 2010). It uses an adapted version of the EAGLES tag set and has encoded tags, such as grade for adjectives. It contains 76,318 unique (lemma, PoS) pairs (13.09% fewer than *aLexiS*).

- The *TIP Conjugator* of Spanish verbs[11] (Carreras-Riudavets et al., 2010). It provides different conjugations accepted by the Spanish academies of several geographical areas, such as in Río de la Plata and Canary Islands. It contains 12,862 unique (lemma, verbs) pairs (9.73% more than *aLexiS*, but only has information for verbs).

- *AnCora-Verb-Es.* A lexicon of Spanish verbs[12] (Aparicio et al., 2008). It has mappings between syntactic functions, arguments and thematic roles for each predicate. Each verbal predicate is related to one or several semantic classes differentiated according to

[10] According to the ELRA website `http://catalog.elra.info/product_info.php?products_id=30`, Nov. 2016.

[11] Available at `http://tip.dis.ulpgc.es/conjugar-verbo/`, Nov. 2016.

[12] Available at `http://clic.ub.edu/ancora/`, Nov. 2016.

four event classes (accomplishments, achievements, states and activities). It contains 1,965 unique (lemma, verbs) pairs (83.08% fewer than *aLexiS*).

	Unique pairs compared to aLexiS
GilcUB-M Dictionary	29.12% less
Freeling	13.09% less
TIP Conjugator	9.73% more (only verbs)
Ancora-Verb-Es	83.08% less

Table 2: *aLexiS* compared to other Spanish lexica in terms of coverage.

Table 2 summarizes the comparison between *aLexiS* and other Spanish lexica in terms of coverage. Table 3 shows the information extracted from the different resources we selected to create *aLexiS*.

According to Molinero et al. (2009), LEFFE contains over 165,000 unique (*lemma*, *tag*) pairs, which correspond to approximately 680,000 unique (*form*, *tag*) pairs. Taking into account that we only extracted some entries (with tags such as noun and adjective), the number of unique (*lemma*, *tag*) pairs pulled out was 101,920, of which 69,879 were tested on the DRAE. This corresponds to 68.56% of the extracted entries. The number of unique (*form*, *tag*) pairs tested on DRAE was 602,393, 88.59% of the total. OSLIN-es contains approximately 115,876 unique (lemma, PoS) pairs and 1,053,401 inflected forms (including homographic forms)[13]. In this case, we only extracted 96,852 unique (*lemma*, *tag*) pairs. This corresponds to 83.58% of the lexicon. Of them, 58,743 pairs were tested on DRAE, that is, 60.65% of the extracted entries.

Table 4 shows the number of lemmas and forms in *aLexiS* classified by lexical category. The vast majority are tagged as nouns ($\sim$ 49,200), representing over 107,000 inflected forms added to *aLexiS*.

As explained in García-Méndez et al. (2018), earlier approaches combined resources of varying formatting quality. Conversely, we chose them for their coverage and accuracy. As shown in Table 2, we were able to collect more lemmas and forms by combining the selected resources than by taking information from them separately (considering only extracted and tested information).

4.2. Experimental text corpus

We evaluated the system manually and automatically. We chose not to apply commonly used measures from the state-of-the-art, such as ROUGE (Lin & Hovy, 2003) and BLEU (Papineni et al., 2002), among others, because they only weakly reflect human judgements of system outputs as generated by end-to-end NLG, as supported by Novikova et al. (2017).

Even though it may be used for general NLG purposes, we focused on a real application, augmentative and alternative communication within the *Accegal* project[14], integrating the system with the *PictoDroid Lite* communicator[15]. First we created a dataset of Spanish sentences and clauses[16]. We discarded the sentences whose complexity exceeded the objectives of our communicator, like those containing subordinate clauses or explanations after a colon such as *Allí estaban sus amigos: el pato, el gato y el pájaro* 'Their friends were there: the duck, the cat and the bird'. Of course our system could generate them, but in separate sentences (one saying who the friends were and other that they were there). We then preprocessed the result to extract the main words within the sentences (nouns, pronouns, proper names, verbs, adjectives and adverbs). Next we lemmatized all these words except for the nouns and pronouns[17]. The resulting dataset had 948 sentences in Spanish and the corresponding main words.

4.3. Evaluation procedure and results

Given a target sentence, we introduced its main words into the automatic NLG system and studied the generated sentences. When the match between the target and generated sentence was total, automatic generation was considered successful. This happened with 736 sentences, 77.64% of the total. The remaining 212 were manually inspected by 5 different annotators, all of them NLP researchers from the *GTI Research Group* in *atlanTTic*, *University of Vigo*. The annotations were chosen from the options in Table 5.

4.3.1. Error type

We considered the six error types in Table 5:

- Morphological error (a): the gender and/or number and/or person features of one or more words of the sentence were not correctly inflected. Sentence 1 in Table 7 is an example, where the verb *coger* 'pick up' is not correctly inflected.

- Syntactic error (b): the sentence had missing elements such as conjunctions and prepositions (sentence 2 in Table 7).

- *aLexiS* error (c): one or more words of the sentence were not in *aLexiS*, so the system treated them as proper names. As a result the inflection and other actions were not correct. Note that if a word is considered a proper name, its first letter is a capital letter (sentence 3 in Table 7).

[13] We thank Maarten Janssen for providing us with this data.

[14] Available at http://www.accegal.org/en/ .

[15] Available at http://www.accegal.org/en/pictodroid-lite/ .

[16] Available at http://www.gti.uvigo.es/index.php/en/resources/6-augmentative-and-alternative-communication-clauses-annotated-dataset-for-natural-language-generation.

[17] If we lemmatized the nouns and pronouns, the system would have no way to know that the user wants to generate a sentence containing a noun or pronoun in plural form, since the features of these words do not depend on other constituents of the sentence, as in the case of adjectives, which depend on the noun that they modify.

	Initial		Extracted		Extracted and Tested			
	(*lem, tag*)	(*form, tag*)	(*lem, tag*)	%lem/Ini	(*lem, tag*)	(*form, tag*)	%lem/Ini	%lem/Ext
LEFFE	165,000	680,000	101,920	61.76%	69,879	602,393	42.35%	68.56%
OSLIN-es	115,876	1,053,401	96,852	83.58%	58,743	778,150	51.63%	60.65%

Table 3: *aLexiS* lemma and form information extraction from LEFFE and *OSLIN-es*.

Category	Lemmas	Forms
Adjective	24,584	82,387
Adverb	2,275	2,275
Conjunction	37	37
Determiner	37	108
Noun	49,206	107,557
Preposition	30	30
Pronoun	32	76
Verb	11,611	649,092
TOTAL	87,812	841,562

Table 4: *aLexiS* size by lexical category.

Feature	Values
Error type	Morphological, syntactic, lexicon, grammar, orthographic, target, lemmatizer
Evaluation	0-5
Best generation	Optional
Generation suggestion	Optional

Table 5: Annotated features.

	Number of sentences
No consensus in best realization	0
No consensus in error with consensus in best realization	24
Positive annotations (total consensus)	188

Table 6: Distribution of the three realization cases of our dataset.

- Grammar error (d): the complexity of the target sentence exceeded the capabilities of our grammar. For example, those without a SVO structure (sentence 4 in Table 7). In this case the system simply repeated the input words.
- Target error (e): the target was not correct (sentence 5 in Table 7).
- Lemmatizer error (f): the lemmatizer did not extract the input words correctly and consequently our system was unable to generate the sentence. For instance, the lemmatizer incorrectly extracted the colour pink in sentence 6 of Table 7 as the subject instead of the proper name *Rosa*.

4.3.2. Rating

The annotators rated the quality of the generation according to the following scale:

- 0: the sentence was not generated. The system simply repeated its main elements, as in sentence 4 in Table 7.
- 1: the meaning of the generated sentence was far from the target (sentence 6 in Table 7).
- 2: the information in the target could be understood from the generated sentence (sentence 1 in Table 7).
- 3: the differences between the target and the generated sentences were minimal and they did not affect the meaning (sentence 2 in Table 7).
- 4: the differences between the target and the generated sentences were limited to punctuation or determiners (sentence 7 in Table 7).
- 5: the target and generated sentences were exactly the same (sentence 8 in Table 7). This rating was assigned automatically in the case of success.

4.3.3. Best generation

If the system offered several possibilities the annotator was requested to choose the best, as for sentence 2 in Table 7, for which the best realization was the third generated sentence.

4.3.4. Suggestion of a result

Inspecting the errors we noticed that most of them were related to SVO ordering and, except for that, the system could have generated the target sentences. The annotators were asked to suggest generation alternatives in that case. A possible suggestion for sentence 5 in Table 7 could be *La sal cayó en el mantel* 'The salt fell on the tablecloth'.

4.3.5. Annotation results

The annotation task took one month. In order to ensure the consistency of the resulting corpus, we provided some guidelines and examples to the annotators in advance. The annotation script produced an XML file with their results. Figure 7 shows an example of a completed annotated sentence.

The final results were summarized as follows. First we distinguished between the cases in which our NLG system returned a single possibility and those with several generated sentences. In the first case we tagged the error type as that indicated by the majority of the annotators. Otherwise, we tagged the sentence with *no consensus in error type*. We calculated the final rating of each generated sentence as the arithmetic average of the ratings by the five annotators. In the second case we first looked for a consensus in the *best_realization* field. If there was no consensus, we tagged the sentence with *no consensus in best realization*. If best realizations were proposed in the second case

ID	Target	System input	Generated sentence/s
1	*Coge el tapón de la botella.* 'Get the stopper of the bottle.'	*coger, tapón, botella* 'get, stopper, bottle'	*Cojo el tapón de la botella.* 'I get the stopper of the bottle.'
2	*La niña escribe en la arena.* 'The girl writes on the sand.'	*niña, escribir, arena* 'girl, write, sand'	*La niña escribe la arena.* *La niña escribe con la arena.* 'The girl writes the sand.' 'The girl writes with the sand.'
3	*Los cerditos ven al lobo.* 'The piglets see the wolf.'	*cerditos, ver, lobo* 'piglets, see, wolf'	*Cerditos ve al lobo.* 'Piglets sees the wolf.'
4	*Cayó sal al mantel.* 'The salt fell on the tablecloth.'	*caer, sal, a, mantel* 'fall, salt, on, tablecloth'	*caer sal a mantel* 'fall salt on tablecloth'
5	*Yo hago pis en el water.* 'I pee in the toilet.'	*hacer, pis, en, water* 'pee, in, toilet'	*Yo hago pis en Water.* 'I pee in Toilet.'
6	*Rosa tiene ropa roja.* 'Rosa has red cloth.'	*rosa, tener, ropa, rojo* 'pink, have got, cloth, red'	*El rosa tiene ropa roja.* 'The pink has pink cloth.'
7	*Mamá corta la barriga del lobo.* 'Mum cuts the belly of the wolf.'	*mamá, cortar, barriga, de, lobo* 'mum, cut, belly, of, wolf'	*La mamá corta la barriga del lobo.* 'The mum cuts the belly of the wolf.'
8	*El papá y el niño pescan con la caña en el río.* 'The dad and the child fish with the fishing rod in the river.'	*papá, niño, pescar, caña, en, río* 'dad, child, fish, fishing rod, in, river'	*El papá y el niño pescan con la caña en el río.* 'The dad and the child fish with the fishing rod in the river.'
9	*El pantalón es morado.* 'The trousers are purple.'	*pantalón, ser, morado* 'trousers, be, purple'	EXACT MATCH
10	*Mamá cepilla al perro.* 'Mum brushes the dog.'	*mamá, cepillar, perro* 'mum, brush, dog'	
11	*El bebé empieza a caminar.* 'The baby starts to walk.'	*bebé, empezar, caminar* 'baby, start, walk'	
12	*Quiero comer melón y limón.* 'I want to eat melon and lemon.'	*querer, comer, melón, limón* 'want, eat, melon, lemon'	
13	*Mamá se seca el pelo con el secador.* 'Mum dries her hair with the dryer.'	*mamá, se, secar, pelo, con, secador* 'mum, dry, hair, with, dryer'	
14	*Las abejas vuelan alrededor de la flor rosa.* 'The bees fly around the pink flower.'	*abejas, volar, alrededor, de, flor, rosa* 'bees, fly, around, of, flower, pink'	
15	*El niño infla un globo gigante de color azul.* 'The child inflates a giant blue ballon.'	*niño, inflar, un, globo, gigante, de, color, azul* 'child, inflate, a, balloon, giant, of, colour, blue'	
16	*El libro y el estuche están dentro de la mochila.* 'The book and the pencil case are inside the schoolbag.'	*libro, estuche, estar, dentro, de, mochila* 'book, pencil case, be, inside, of, schoolbag'	
17	*Los niños pintan con un lápiz azul en papel blanco.* 'The children paint with a blue pencil on the white paper.'	*niños, pintar, un, lápiz, azul, en, papel, blanco* 'children, paint, a, pencil, blue, on, paper, white'	

Table 7: Example of sentences that were generated automatically by our system, compared with the targets.


```
-<TAGGING>
  -<CLAUSE>
     <TARGET>Papá oso coge el plato grande</TARGET>
    -<Generated_Clauses>
      -<Clause>
         El papá del oso coge el plato grande.
         <Error>b</Error>
         <Rating>1</Rating>
       </Clause>
      -<Clause>
         El papá y el oso cogen el plato grande.
         <Error>b</Error>
         <Rating>1</Rating>
       </Clause>
      -<Clause>
         El papá oso coger el plato grande.
         <Error>a</Error>
         <Rating>2</Rating>
       </Clause>
     </Generated_Clauses>
     <Best_realisation>3</Best_realisation>
     <Suggestion_for_Generation>EMPTY</Suggestion_for_Generation>
   </CLAUSE>
 </TAGGING>
```


Figure 7: Annotation example.

and there was consensus among the annotators, we tagged and rated the best candidate generated by the system as in the first case. Table 6 shows the distribution of the cases for our dataset.

When the annotators agreed in the second case in best realization and error, the average rating of the annotated sentences was 2. This also happened when there was no consensus in error but the annotators coincided in the best realization. The annotators made 238 different generation suggestions, of which 89 were correctly generated by our library (37.39%).

Note that the tests covered various features of Spanish grammar such as different types of sentences (affirmative, negative, interrogative, coordinate, passive, etc.), the entire Spanish verb conjugation and constructions with different categories of words (adjectives, nouns, pronouns, etc.). Table 7 shows some examples of generated sentences. The first eight correspond to failures of our system that we have used as examples in this section. The rest are correct generations that illustrate the functionality of the system.

4.4. Agreement measures

Manual evaluation was assessed with two well-known agreement measures to obtain a robust estimate of the differences between the annotators: Krippendorff's *Alpha*-reliability and accuracy.

Krippendorff's *Alpha*-reliability (expression 1) (Krippendorff, 2012, 2011) is a reliability coefficient that measures agreement among observers. It analyses whether the resulting data can be trusted to represent something real. Unlike other specialized coefficients, *Alpha* is a generalization of several known reliability indices. It allows researchers to judge a variety of data with the same reliability standard. It is valid for any number of annotators, can be applied to different variable types and metrics (e.g. nominal, ordered, interval, etc.) and can handle small or large sample sizes and incomplete/missing data.

$$Alpha = 1 - \frac{D_o}{D_e} \quad (1)$$

where D_o is the observed disagreement between the annotators and D_e is the disagreement expected by chance rather than attributable to the properties of the coded units. When the annotators agree perfectly $Alpha = 1$, and when their agreement level seems by chance $Alpha = 0$. For the data generated by any method to be reliable, *Alpha* should be far from this extreme, ideally $Alpha = 1$. Expressions 2 and 3 define the two disagreement measures.

$$D_o = \frac{1}{n} \sum_c \sum_k o_{ck} \cdot \delta^2_{c,k} \quad (2)$$

$$D_e = \frac{1}{n(n-1)} \sum_c \sum_k n_c \cdot n_k \cdot \delta^2_{c,k} \quad (3)$$

where entities o_{ck} , n_c, n_k and n refer to the frequencies of values in coincidence matrices. The first is calculated as follows:

$$o_{ck} = \sum_n \frac{\textit{Number of c - k pairs in sentence u}}{\textit{Number of annotators-1}} \quad (4)$$

	k	$\sum$
c	. . . . . o_{ck}	. n_c
$\sum$	n_k	n

Table 8: Coincidence matrix from two different annotators into a $k \times k$ square matrix.

Expression 5 defines the difference function δ.

$$\delta_{c,k} = \begin{cases} 0 & if \quad c = k \\ 1 & if \quad c \neq k \end{cases} \quad (5)$$

Our evaluation scenario focused on nominal data because we measured the agreement in errors (a, b, c, d, e, f) of five observers with no missing data from our dataset. The first step was to build a reliability data matrix, a 5-observers-by-229-sentences matrix[18], containing 5×229 values (c and k respectively).

Reliability data matrix		**Sentences**						
		1	**2**	**3**	**...**	**125**	**...**	**229**
Observers	**Annot. 1**	b	a	d	...	c	...	d
	Annot. 2	b	a	d	...	c	...	d
	Annot. 3	b	a	a	...	e	...	d
	Annot. 4	b	a	a	...	f	...	d
	Annot. 5	b	a	a	...	c	...	d

Table 9: Reliability data matrix of our annotated dataset.

The second step was to tabulate coincidence in units (Table 10). Coincidence matrices account for all values contained in a reliability data matrix. They differ from the familiar contingency matrices, which account for units in two dimensions, not values. Our coincidence matrix tabulated all pairable errors from the five different annotators into a 6-by-6 square matrix. A coincidence matrix omits references to annotators. It is symmetric with respect to its diagonal, which contains all the perfect matches. Note that coincidences are counted twice in the coincidence matrix. Disagreements (represented by off-diagonal cells) are also counted twice, but in different cells. Table 10 shows the form of our coincidence matrix.

Errors	**a**	**b**	**...**	**f**
a	199.6	7.6	...	0.8
b	7.6	182.4	...	2.8
...	...	...	...	...
f	0.8	2.8	...	11.2

Table 10: Coincidence matrix of our annotated dataset.

Agreement measure	**Value**
Alpha	0.598
Accuracy	0.689

Table 11: Overall inter-annotator agreement considering the five annotators.

We next estimated the agreement between pairs of annotators with the accuracy indicator. This is defined in terms of the observed disagreement D_o, as shown in Equation 6:

$$Accuracy = 1 - D_o \quad (6)$$

The accuracy is simply the average of the proportions given by the diagonal elements of the coincidence matrix. Note that it neither accounts for (dis)agreement by chance nor for the ordering of possible values. Table 11 shows

[18]Our system generated 229 sentences for the 212 target sentences in the corpus because there were several generated candidates for some targets.

		Observers				
		Annot. 1	Annot. 2	Annot. 3	Annot. 4	Annot. 5
Observers	Annot. 1	-	**0.755**	0.501	0.564	**0.646**
	Annot. 2		-	0.575	0.570	**0.602**
	Annot. 3			-	**0.616**	0.578
	Annot. 4				-	0.561
	Annot. 5					-

Table 12: *Alpha* reliability between pairs of annotators.

		Observers				
		Annot. 1	Annot. 2	Annot. 3	Annot. 4	Annot. 5
Observers	Annot. 1	-	**0.812**	0.616	0.668	**0.734**
	Annot. 2		-	0.664	0.664	**0.690**
	Annot. 3			-	**0.707**	0.672
	Annot. 4				-	0.664
	Annot. 5					-

Table 13: Accuracy measures between pairs of annotators.

promising overall *Alpha* and accuracy results, which are even better in Tables 12 and 13, representing agreement by pairs of annotators (consider as a reference the inter-agreement measures in Dorussen et al. (2005), Poesio & Artstein (2005) and Pestian et al. (2012), for example).

4.5. Comparison with the automatic SimpleNLG version

We are not aware of any other system or library that performs NLG automatically. Therefore, we took the enhanced (automatic) Spanish version of our SimpleNLG library as a reference.

We first built a manual Spanish version of the SimpleNLG library by writing new code to satisfy the linguistic requirements of Spanish. This adaptation also uses the complete and reliable *aLexiS* lexicon with Spanish morphology as the basis to generate sentences. The enhanced automatic version uses Elsa, which contains not only morphological data (like *aLexiS*) but also syntactic and semantic information (because this version does not use a grammar).

In the manual version, to generate a sentence it is first necessary to create the syntagms that compose it and then join them paying attention to their syntactic and semantic function. In order to generate the sentence *El lobo come a la abuela* 'The wolf eats the grandmother', we have to create the nominal syntagm *la abuela* 'the grandmother' and integrate it into a prepositional syntagm *a la abuela* 'to the grandmother'. At the same time, it is also necessary to construct the subject of the sentence *el lobo* 'the wolf' and the predicate with the main verb *comer* 'eat'. Finally, we need to integrate the subject and the predicate with the prepositional complement in the final sentence.

The enhanced version manages all these stages automatically. It follows a hybrid approach that combines the knowledge-base of Elsa with a language model, according to a statistical approach, to infer prepositions. Together with the lexical rules in the adapted library and those we implemented, the enhanced version can generate coherent and complete sentences. It inserts conjunctions, determiners and prepositions automatically. In addition, if the input words contain a time adverb, like *mañana* 'tomorrow', the verbal tense of the sentence is automatically adjusted. In the special case of verbs that can be used reflexively or non-reflexively, the system generates the sentence depending on the corresponding probabilities. The system gets this information from the language model we also created. We developed an algorithm to construct the sentence word by word based on the linguistic knowledge in the lexicon and the grammar rules that we implemented for Spanish. The algorithm relies on the morphological categories of the words and infers their possible syntactic function within the sentence by also using their semantic data. We refer the reader to García-Méndez et al. (2018) for more detail.

We compared the new library proposed in this paper with the automatic Spanish version of SimpleNLG that we created using the dataset in Section 4.2. Tables 14 and 15 show the comparison. The new NLG library outperformed the automatic version of SimpleNLG. The former generate 77.64% of the dataset sentences, but the latter only succeeded in generating 38.29% of them. Besides, the new library generated 390 sentences that our automatic version of SimpleNLG was unable to create. The automatic version of SimpleNLG was better for only 17 sentences.

The performance gap seems due to the difficulty to introduce new realizations in the automatic SimpleNLG version, since these must be codified within the library. However, using our new NLG library and a grammar, a new realization is a new linguistic tree that is independent from the code of the library. This explains why, for many target sentences, the output of our automatic SimpleNLG version is considered incorrect due to the presence of an article, while the NLG library generates several variants for a single target.

Target	**Using our NLG system**	**Using our version of *SimpleNLG***
El niño pasea por la calle. 'The children walk through the street.'	*El niño pasea en la calle.* 'The children walk on the street.'	*El niño pasea la calle.* 'The children walk the street.'
Corté el filete con tijeras. 'I cut the steak with scissors.'	*Corto el filete con tijeras.* 'I cut the steak with scissors.' (in present and with elided subject)	*Yo corto el filete con las tijeras.* 'I cut the steak with the scissors.'
Mamá se seca el pelo con el secador. 'Mum dries her hair with the dryer.'	*Mamá se seca el pelo con el secador.* 'Mum dries her hair with the dryer.'	*La mamá y se secan el pelo con el secador.* 'Mum dries her hair with the dryer.'
Me gusta la pandereta. 'I like the tambourine.'	*Me gusto la pandereta.* 'I like the tambourine.'	*Me gusto la pandereta.* 'I like the tambourine.'
El lobo feroz siguió a caperucita. 'The big bad wolf followed Little Red Riding Hood.'	*El lobo feroz sigue a Caperucita.* 'The big bad wolf follows Little Red Riding Hood.'	*El lobo y feroz siguen el caperucita.* 'The wolf and fierce follow the Little Red Riding Hood.'
Los cerditos corren rápido. 'The piglets run quickly.'	*Cerditos corre rápido.* 'Piglets runs quickly.'	*Los cerditos corren rápidos.* 'The piglets run quickly.'
La bruja envenena la manzana. 'The witch poisons the apple.'	*La bruja envenena la manzana.* 'The witch poisons the apple.'	*La bruja, envenenar y la manzana.* 'The witch, poison and the apple.'
El libro y el estuche están dentro de la mochila. 'The book and the pencil case are inside the school bag.'	*El libro y el estuche están dentro de la mochila.* 'The book and the pencil case are inside the school bag.'	*El libro está dentro de la mochila.* 'The book is inside the school bag.'

Table 14: Comparison between our new NLG system and the automatic Spanish version of *SimpleNLG* - examples.

		Enhanced *SimpleNLG*		
		Correct	**Incorrect**	**Total**
Our library	**Correct**	346	390	736
	Incorrect	17	195	212
	Total	363	585	948

Table 15: Comparison between our new NLG system and the automatic Spanish version of *SimpleNLG* - automatic generation success.

5. Conclusions

Spanish NLG is a promising field where automation and flexibility deserve attention. We have described an alternative to the common NLG practice of using templates in some steps of the generation process (Ramos-Soto et al., 2016). Relying on the *aLexiS* lexicon and our grammar, our system performs fully automatic NLG from a minimum set of words. Additional lexical resources can be easily integrated thanks to the automatic building process. The architecture allows domain-dependent components to be separated from domain-independent components, which can be reused or substituted. In this regard, the *aLexiS* lexicon and the grammar are interesting results in themselves that could be useful for many other NLG systems or applications. Besides, the system could also be used to check for grammatical errors.

To the best of our knowledge this is the first attempt to create a fully automatic system for Spanish NLG following a hybrid approach, that is, combining a knowledge-based approach (linguistic data, i.e. vocabulary and grammar realizations) and a machine learning approach (language model). Our system is able to generate complete, coherent and correctly spelled sentences from the main word sets supplied by the user. Noteworthy is the easy extension of our system to other languages due to its modular design and implementation, focused on integrability, portability and efficiency for feasible integration in a wide range of digital devices.

It may be argued that modularity is compromised by the vocabulary and the grammar required to perform the NLG task in an automatic way. Indeed, the system relies on this data to take most of the surface realization decisions. For this reason, we have provided a detailed explanation of the lexicon creation procedure and the grammar used. We are aware of the fact that our library expects the user to introduce the input words respecting an SVO order (broadly tolerated in the NLG field). In this regard, a future line of research is predicting the best grammar realization for a set of non-ordered input words to further improve the performance of our system.

Automatic and manual evaluation resulted in a positive experimental outcome. We applied state-of-the-art metrics as well as human supervision to assess system performance. The system was able to automatically and correctly generate 77.64% of the sentences. Due to the novelty of our automatic approach and the lack of datasets for similar tasks, we had to create an automatic version of other library from the literature as a competitor (see Section 4.5). According to this comparison, we obtained a significant 35.35% accuracy improvement. The evaluations of other state-of-the-art NLG solutions seek an exact match between the target and generated texts. Conversely, we performed an insightful analysis of similarities and main differences at a semantic level. In the AAC use case, our system may help people with communication impairments to express themselves in an easier, faster and more natural way.

As future research lines, first we will adapt our system to use by other languages. Second, we will study to what extent our system may help people with communication disabilities. Last but not least, as previously said, we seek to predict the best grammar realization for a set of non-ordered input words to further improve performance.

Acknowledgements

This work was partially supported by Mineco grant TEC2016-76465-C2-2-R and Xunta de Galicia grants GRC 2014/046 and ED341D R2016/012.

References

Andersen (2016). Fairy tales of Hans Christian Andersen (Spanish). Available 28/11/2016 at https://es.wikisource.org/wiki/Cuentos_clásicos_ para_niños.

Anonymous (2016). Traditional Spanish tales. Available 28/11/2016 at http://loscuentostradicionales.blogspot.com.es.

Aparicio, J., Taulé, M., & Mart, M. (2008). Ancora-verb: A lexical resource for the semantic annotation of corpora. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, & D. Tapias (Eds.), *Proc. of the 6th International Conference on LREC*. Marrakech, Morocco: ELRA.

Appelt, D. E. (1985). *Planning English Sentences*. New York, NY, USA: Cambridge University Press.

Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., & Padró, M. (2006). Freeling 1.3: Syntactic and semantic services in an open-source NLP library. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006)*. Genoa, Italy: ELRA.

Barzilay, R., & Lapata, M. (2008). Modeling local coherence: An entity-based approach. *Computational Linguistics*, *34*, 1–34.

Bateman, J. A. (1997). Enabling technology for multilingual natural language generation: The KPML development environment. *Nat. Lang. Eng.*, *3*, 15–55. URL: http://dx.doi.org/10.1017/S1351324997001514. doi:10.1017/S1351324997001514.

Bel, N., Padró, M., & Necsulescu, S. (2011). A method towards the fully automatic merging of lexical resources. In *Proc. of Workshop on Language Resources, Technology and Services in the Sharing Paradigm* (pp. 8–15). ACL.

Belz, A. (2008). Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Nat. Lang. Eng.*, *14*, 431–455. URL: http://dx.doi.org/10.1017/S1351324907004664. doi:10.1017/S1351324907004664.

Cahill, A., & Van Genabith, J. (2006). Robust pcfg-based generation using automatically acquired lfg approximations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* (pp. 1033–1040). Association for Computational Linguistics.

Carreras-Riudavets, F. J., Hernández-Figueroa, Z., & Rodríguez Rodríguez, G. (2010). *La conjugación de verbos en español y su morfología*. LULU.

Chen, J., Bangalore, S., Rambow, O., & Walker, M. A. (2002). Towards automatic generation of natural language generation systems. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1* (pp. 1–7). Association for Computational Linguistics.

Cheyer, A., & Guzzoni, D. (2007). Method and apparatus for building an intelligent automated assistant. URL: https://www.google.com/patents/US20070100790 uS Patent App. 11/518,292.

Crouch, D., & King, T. H. (2005). Unifying lexical resources. In *Proc. of Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes*.

Dale, R. (1989). Cooking up referring expressions. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics* (pp. 68–75). Association for Computational Linguistics.

Dale, R. (1992). *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes. ACL-MIT Series in Natural Language Processing*. MIT Press.

Dale, R., Geldof, S., Prost, J.-P. et al. (2005). Using natural language generation in automatic route description. *Journal of Research and practice in Information Technology*, *37*, 89.

Dale, R., & Reiter, E. (1995). Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*, *19*, 233–263.

Danlos, L., & Sagot, B. (2008). Constructions pronominales dans Dicovalence et le lexique-grammaire–intégration dans le Lefff. In *Proc. of the 27th Lexicon-Grammar Conference*. L'Aquila, Italy. URL: https://hal.inria.fr/inria-00524741.

Dorussen, H., Lenz, H., & Blavoukos, S. (2005). Assessing the reliability and validity of expert interviews. *European Union Politics*, *6*, 315–337.

Dušek, O., & Jurcicek, F. (2015). Training a natural language generator from unaligned data. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 451–461). volume 1.

Fiedler, A. (2005). Natural language proof explanation. In *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday* (pp. 342–363). URL: https://doi.org/10.1007/978-3-540-32254-2_20. doi:10.1007/978-3-540-32254-2_20.

Frank, M. C., Goodman, N. D., & Tenenbaum, J. B. (2009). Using speakers' referential intentions to model early cross-situational word learning. *Psychological science*, *20*, 578–585.

García-Méndez, S., Fernández-Gavilanes, M., Costa-Montenegro, E., Juncal-Martínez, J., & González-Castaño, F. J. (2018). Automatic Natural Language Generation Applied to Alternative and Augmentative Communication for Online Video Content Services using SimpleNLG for Spanish. In *Proceedings of the 15th Web for All Conference on The Future of Accessible Work*. ACM. Accepted, waiting for publication.

Gardent, C., & Narayan, S. (2015). Multiple adjunction in feature-based tree-adjoining grammar. *Computational Linguistics*, *41*, 41–70.

Gavilanes, M. F. (2012). *Adquisición y representación del conocimiento mediante procesamiento del lenguaje natural*. Ph.D. thesis Universidade da Coruña.

Gervás, P. (2001). An expert system for the composition of formal spanish poetry. In *Applications and Innovations in Intelligent Systems VIII: Proceedings of ES2000, the Twentieth SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence, Cambridge, December 2000* (pp. 19–32). London: Springer London. URL: https://doi.org/10.1007/978-1-4471-0275-5_2. doi:10.1007/978-1-4471-0275-5_2.

Goldberg, E., Driedger, N., & Kittredge, R. I. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert: Intelligent Systems and Their Applications*, *9*, 45–53. URL: https://doi.org/10.1109/64.294135. doi:10.1109/64.294135.

Grimm (2016). Grimm's fairy tales (Spanish). Available 28/11/2016 at http://www.grimmstories.com/es.

Grishman, R., & Sandoval, A. M. (1991). *Introducción a la lingüística computacional*. Visor.

Hovy, E. H. (1993). Automated discourse generation using discourse structure relations. *Artificial Intelligence*, *63*, 341 – 385. URL: http://www.sciencedirect.com/science/article/pii/0004370293900213. doi:https://doi.org/10.1016/0004-3702(93)90021-3.

Janssen, M. (2005). Open source lexical information network. In *Proc. of the 3rd International Workshop on Generative Approaches to the Lexicon* (pp. 400–410).

Janssen, M. (2009). Lexical functions vs. inflectional functions. In *Proc. 4th International Conference on Meaning-Text Theory* (pp. 189–198).

Koller, A., Striegnitz, K., Gargett, A., Byron, D., Cassell, J., Dale, R., Moore, J., & Oberlander, J. (2010). Report on the second NLG challenge on generating instructions in virtual environments (GIVE-2). In *Proceedings of the 6th International Natural Language Generation Conference* INLG '10 (pp. 243–250). Stroudsburg, PA, USA: Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=1873738.1873776.

Krippendorff, K. (2011). Computing krippendorff's alpha-reliability.

Krippendorff, K. (2012). *Content analysis: An introduction to its* *methodology*. Sage.

Kutlak, R., Mellish, C., & van Deemter, K. (2013). Content selection challenge - University of Aberdeen entry. In *Proceedings of the 14th European Workshop on Natural Language Generation* (pp. 208–209). Sofia, Bulgaria: Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/W13-2133.

Lampouras, G., & Vlachos, A. (2016). Imitation learning for language generation from unaligned data. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 1101–1112).

Langkilde, I., & Knight, K. (2002). Halogen statistical sentence generator. In *Proceedings of the Association for Computational Linguistics'02 Demonstrations Session, Philadelphia* (pp. 102–103).

Lim-Cheng, N. R., G. Fabia, G. I., G. Quebral, M. E., & Yu, T. (2014). Shed: An online diet counselling system. In *DLSU Research Congress*.

Lin, C.-Y., & Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 71–78). Association for Computational Linguistics.

Liu, W. P., Georgescu, B., Zhou, S. K., & Comaniciu, D. (2017). Automatic generation of radiology reports from images and automatic rule out of images without findings. US Patent App. 15/158,375.

Maggiori, E. (2013). Desarrollo de una gramática para aserciones simples en español y su implementación en prolog. In *XVI Concurso de Trabajos Estudiantiles (EST)-JAIIO 42 (2013)*.

Mairesse, F., & Walker, M. (2007). Personage: Personality generation for dialogue. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (pp. 496–503).

Mann, W. (1982). Text generation. *Comput. Linguist.*, *8*, 62–69. URL: http://dl.acm.org/citation.cfm?id=972932.972935.

McKeown, K. R. (1985). *Text generation: using discourse strategies and focus constraints to generate natural language text*. Cambridge University Press Cambridge [Cambridgeshire] ; New York. URL: http://www.loc.gov/catdir/enhancements/fy1001/84019889-t.html.

Mei, H., Bansal, M., & Walter, M. R. (2015). What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *arXiv preprint arXiv:1509.00838*, .

Mirkovic, D., & Cavedon, L. (2008). Dialogue management using scripts. URL: https://encrypted.google.com/patents/EP1891625A1?cl=und eP Patent App. EP20,060,759,358.

Molinero, M. A., Sagot, B., & Nicolas, L. (2009). A morphological and syntactic wide-coverage lexicon for Spanish: The Leffe. In *RANLP*. Borovets, Bulgaria. URL: https://hal.inria.fr/inria-00616693.

Nakanishi, H., Miyao, Y., & Tsujii, J. (2005). Probabilistic models for disambiguation of an hpsg-based chart generator. In *Proceedings of the Ninth International Workshop on Parsing Technology* (pp. 93–102). Association for Computational Linguistics.

Necsulescu, S., Bel Rafecas, N., Padró, M., Marimon, M., & Revilla, E. (2011). Towards the automatic merging of language resources. In *First International Workshop on Lexical Resources: an ESSLLI 2011 Workshop; 2011 Aug 1-5; Ljubljana, SI. Ljubljana: ESSLLI; 2011. p. 70-77.* ESSLLI.

Novikova, J., Dušek, O., Curry, A. C., & Rieser, V. (2017). Why we need new evaluation metrics for nlg. *arXiv preprint arXiv:1707.06875*, .

Padró, L., Collado, M., Reese, S., Lloberes, M., & Castellón, I. (2010). Freeling 2.1: Five years of open-source language processing tools. In *Proc. of the 7th International Conference on LREC, 17-23, Valletta, Malta*. URL: http://www.lrec-conf.org/proceedings/lrec2010/summaries/14.html.

Padró, L., & Stanilovsky, E. (2012). Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*. Istanbul, Turkey: ELRA.

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

Peinado, F., Gervs, P., & Daz-agudo, B. (2004). A description logic ontology for fairy tale generation. In *In Forth Int. Conf. on Language Resources and Evaluation: Workshop on Language Resources for Linguistic Creativity* (pp. 56–61).

Pestian, J. P., Matykiewicz, P., Linn-Gust, M., South, B., Uzuner, O., Wiebe, J., Cohen, K. B., Hurdle, J., & Brew, C. (2012). Sentiment analysis of suicide notes: A shared task. *Biomedical informatics insights*, *5*, BII–S9042.

Poesio, M., & Artstein, R. (2005). The reliability of anaphoric annotation, reconsidered: Taking ambiguity into account. In *Proceedings of the workshop on frontiers in corpus annotations ii: Pie in the sky* (pp. 76–83). Association for Computational Linguistics.

Ramos-Soto, A., Bugarín, A., & Barro, S. (2016). On the role of linguistic descriptions of data in the building of natural language generation systems. *Fuzzy Sets Syst.*, *285*, 31–51. URL: https://doi.org/10.1016/j.fss.2015.06.019. doi:10.1016/j.fss.2015.06.019.

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Nat. Lang. Eng.*, *3*, 57–87. URL: http://dx.doi.org/10.1017/S1351324997001502. doi:10.1017/S1351324997001502.

Reiter, E., & Dale, R. (2000). *Building Natural Language Generation Systems*. New York, NY, USA: Cambridge University Press.

Reiter, E., Mellish, C., & Levine, J. (1995). Automatic-generation of technical documentation. *Applied Artificial Intelligence*, *9*, 259–287.

Reiter, E., Robertson, R., & Osman, L. (2003). Lessons from a Failure: Generating Tailored Smoking Cessation Letters. *Artificial Intelligence*, *144*, 41–58. URL: http://www.csd.abdn.ac.uk/~{}ereiter/papers/aij03.pdf.

Reiter, E., Sripada, S., Hunter, J., & Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, *167*, 137–169.

Ruwet, N., Bombín, E., & Hernández, M. M. (1974). *Introducción a la gramática generativa*. Gredos.

Sager, N. (1967). Syntactic analysis of natural language. *Advances in Computers*, *8*, 153–188. URL: https://doi.org/10.1016/S0065-2458(08)60696-8. doi:10.1016/S0065-2458(08)60696-8.

Sagot, B. (2010). The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, & D. Tapias (Eds.), *Proc. of the 7th International Conference on LREC*. Valletta, Malta: ELRA.

Sauper, C., & Barzilay, R. (2009). Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1* ACL '09 (pp. 208–216). Stroudsburg, PA, USA: Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id=1687878.1687909.

Schank, R. C. (1975). *Conceptual Information Processing*. New York, NY, USA: Elsevier Science Inc.

Stent, A., Marge, M., & Singhai, M. (2005). Evaluating evaluation methods for generation in the presence of variation. In *Computational Linguistics and Intelligent Text Processing: 6th International Conference, CICLing 2005, Mexico City, Mexico, February 13-19, 2005. Proceedings* (pp. 341–351). Berlin, Heidelberg: Springer Berlin Heidelberg. URL: https://doi.org/10.1007/978-3-540-30586-6_38. doi:10.1007/978-3-540-30586-6_38.

Tarjan, R. (1971). Depth-first search and linear graph algorithms. In *Switching and Automata Theory, 1971., 12th Annual Symposium on* (pp. 114–121). IEEE.

Thomason, J., Venugopalan, S., Guadarrama, S., Saenko, K., & Mooney, R. (2014). Integrating language and vision to generate natural language descriptions of videos in the wild. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)* (pp. 1218–1227). Dublin, Ireland. URL: http://www.cs.utexas.edu/users/ai-lab/pub-view.php?

PubID=127457.

Wanner, L., Bohnet, B., Bouayad-Agha, N., Lareau, F., & Nicklaß, D. (2010). Marquis: Generation of user-tailored multilingual air quality bulletins. *Applied Artificial Intelligence*, *24*, 914–952.

White, M., Clark, R. A. J., & Moore, J. D. (2010). Generating tailored, comparative descriptions with contextually appropriate intonation. *Computational Linguistics*, *36*, 159–201.

Williams, S., & Reiter, E. (2008). Generating basic skills reports for low-skilled readers. *Natural Language Engineering*, *14*, 495–525.