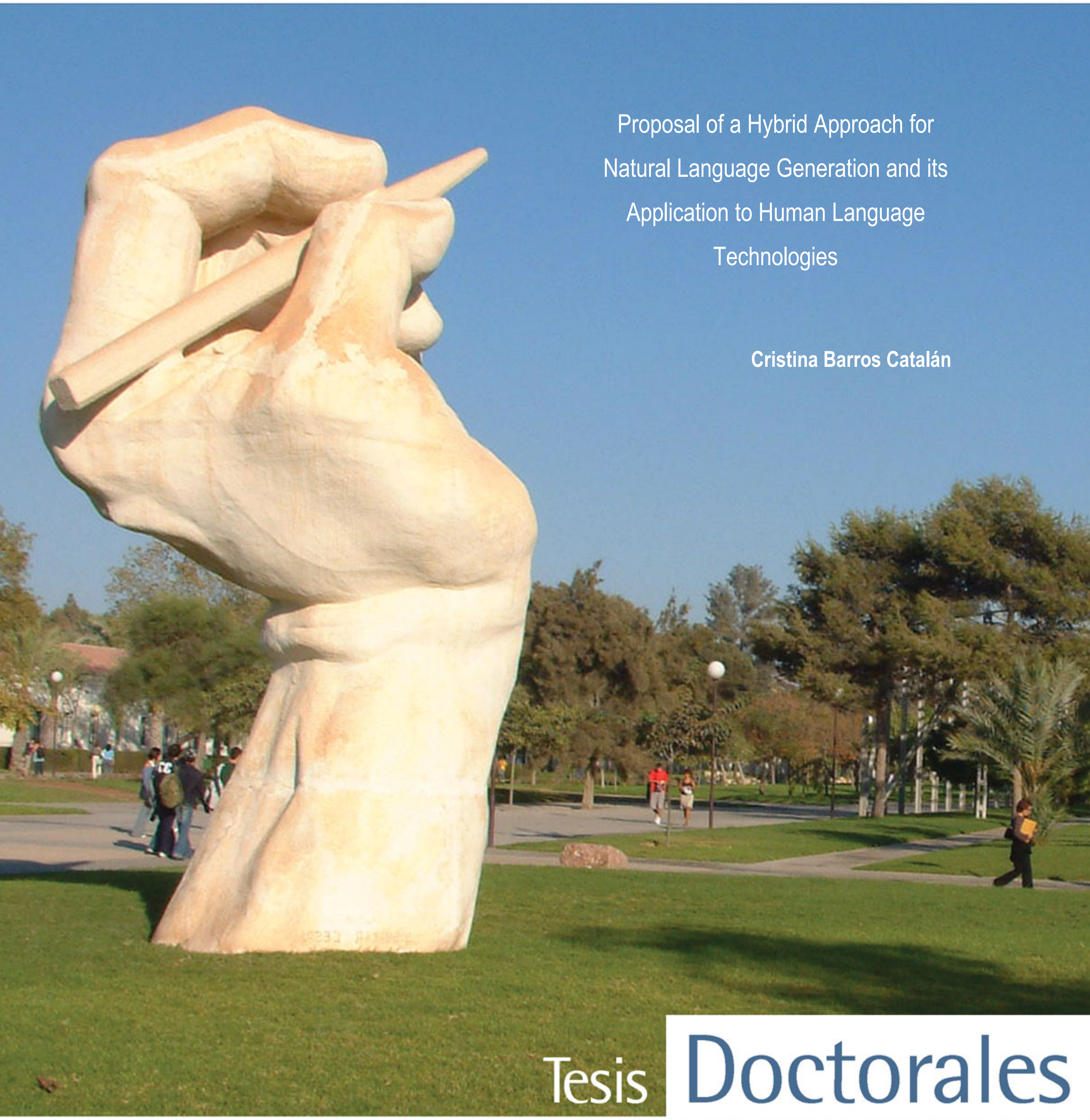


Proposal of a Hybrid Approach for Natural Language Generation and its Application to Human Language Technologies

Cristina Barros Catalán

Universitat d'Alacant
Universidad de Alicante

Departamento de Lenguajes y Sistemas Informáticos
Escuela Politécnica Superior

Proposal of a Hybrid Approach for Natural Language Generation and its Application to Human Language Technologies

Cristina Barros Catalán

Tesis presentada para aspirar al grado de

DOCTOR POR LA UNIVERSIDAD DE ALICANTE

MENCIÓN DE DOCTOR INTERNACIONAL

DOCTORADO EN INFORMÁTICA

Dirigida por

Dr. Elena Lloret Pastor

Esta tesis ha sido financiada por el Gobierno Valenciano a través del proyecto DIIM 2.0: Desarrollo de técnicas Inteligentes e Intractivas de Minería y generación de información sobre la web 2.0 (PROMETEOII/2014/001).

A mi madre, por enseñarme que por muy mal que
vayan las cosas siempre hay solución
A Marimar e Inma, por su apoyo incondicional
A Jose, por ser el pilar fundamental de mi vida que
hace que no caiga y pueda seguir adelante

Acknowledgements

Esta tesis es el fruto de un trabajo de investigación cuyos inicios se remontan al 14 de octubre de 2014, día que entré a formar parte del Grupo de Procesamiento del Lenguaje Natural y Sistemas de Información de la Universidad de Alicante. Antes de ese día nunca me habría planteado vivir este tipo de experiencia, pero no me arrepiento de la decisión que tomé. Estos últimos cinco años han sido un no parar con diferentes retos y experiencias que me han hecho crecer como persona. Ya sea en la participación de congresos, cursos o estancias, la realización de esta tesis me ha permitido ampliar horizontes y conocer diferentes culturas así como cambiar mi forma de ver el mundo. Nunca había sido una persona viajera pero gracias a tesis he tenido la suerte de recorrer el mundo visitando distintos países como Reino Unido, Portugal, Francia, Dinamarca, México o Japón (uno de los países que siempre había querido visitar).

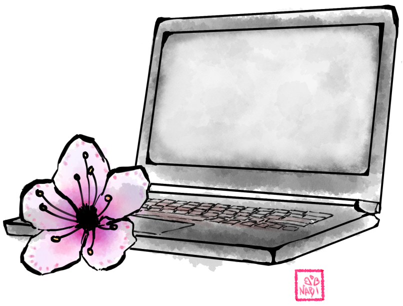
Algo que siempre se tiene que tener en cuenta cuando se desarrolla una tesis doctoral es que no es un camino de rosas. Detrás de cada tesis doctoral hay un gran esfuerzo y constancia en el trabajo realizado. El desarrollo de esta tesis ha sido una manera de retarme a mí misma, ya que, a pesar de mi tartamudez, miedo escénico y que en muchas ocasiones me cuesta expresarme; la tesis ha llegado a "buen puerto". Han habido muchas personas que me han ayudado a abordar este reto y me han animado a seguir adelante cuando pensaba que no podía más. Por esta razón, a todas ellas, les quiero agradecer todos los apoyos que me han transmitido. En especial me gustaría agradecer:

- A Elena Lloret, mi directora de tesis, por tu apoyo, por animarme siempre a seguir adelante y confiar en mí en todo momento. Por todos tus consejos, ideas y opiniones. Por ser positiva cuando yo siempre lo veía todo negro. Por ayudarme a abrir nuevos horizontes y a ver las cosas distintas. Por todo esto y muchísimas cosas más. ¡Gracias de todo corazón!

- A Paloma Moreda y Patricio Martínez por introducirme en el mundo de la investigación, por vuestro apoyo y por toda la ayuda que me habéis dado a lo largo del desarrollo de esta tesis.

- Al Grupo de investigación de Procesamiento del Lenguaje Natural y Sistemas de Información, al Departamento de Lenguajes y Sistemas Informáticos y a la Universidad de Alicante, que me han permitido la realización de esta tesis doctoral.

- A todos mis compañeros de laboratorio, tanto los que han estado antes y los que están ahora, que han estado siempre ahí y me han ayudado muchísimo. En especial me gustaría agradecer a María de los Angeles Herrero por todas las tonterías gatunas que hemos compartido y por los ánimos constantes; y a Lea Canales, por toda su ayuda y por ser paciente al preguntarle dudas sobre la tesis y sus trámites burocráticos.
- A Marta Vicente, por haber sido mi compañera y amiga durante todo el doctorado y la carrera. Por todos tus ánimos, postales y aventuras juntas. ¡Millones de gracias!
- A Laura Haide Pérez y Marina Litvak, porque sus comentarios han servido para mejorar esta tesis.
- A la Dra Dimitra Gkatzia por haberme dejado pertenecer a su grupo de investigación en la Edinburgh Napier University y poder así mejorar la calidad científica de la investigación.
- A Carlos y Lucía, por todas esas tardes de vicio y rock 'n' roll.
- A mi "prima" Inma, por todo el apoyo y cariño que me has dado desde que nos conocemos.
- A Marimar, mi mejor amiga y uno de los pilares de mi vida. Gracias por aguantarme y estar conmigo tanto en los buenos como malos momentos. Aunque estemos lejos, te quiero muchísimo y siempre serás una parte importante de mi vida.
- A mis padres Magdalena y Manuel. Mama, gracias por ser mi modelo a seguir, por ver siempre el lado positivo de las cosas y por ayudarme en caa aspecto de mi vida. Papa, aunque ya no estés aquí te doy las gracias por haber hecho de mí la persona que soy ahora.
- Y por último, a mi mejor amigo, compañero y el amor de mi vida, a Jose. Te podría dar las gracias por un millón de cosas y nunca serían suficientes. Pero lo más es importante y lo que de verdad quiero agradecerte es por haber permanecido a mi lado en cada momento, por apoyarme en todo y por comprenderme cuando a veces no me comprendo ni yo misma.



Contents

List of Figures

List of Tables

Acronyms

ACL	Association for Computational Linguistics
AI	Artificial Intelligence
COLING	International Conference on Computational Linguistics
CC-NLG	Workshop on Computational Creativity in Natural Language Generation
CLN	Comprensión del Lenguaje Natural
D2T	data-to-text
DUC	Document Understanding Conferences
ENLG	European Natural Language Generation Workshop
EACL	European Chapter of the Association for Computational Linguistics
EMNLP	Conference on Empirical Methods in Natural Language Processing
FLM	Factored Language Models
GRE	Generation of Referring Expressions
GLN	Generación del Lenguaje Natural
HanaNLG	Hybrid surfAce realisatioN Approach for Natural Language Generation
INLG	International Natural Language Generation Conference
LM	Language Models
LDA	Latent Dirichlet Allocation
MTT	Meaning-Text Theory
MLF	Modelos de Lenguaje Factorizados

NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
NAACL	North American Chapter of the Association for Computational Linguistics
NE	Named Entities
OTS	Open Text Summarizer
POS	Part-Of-Speech
PLN	Procesamiento del Lenguaje Natural
RST	Rhetorical Structure Theory
SFG	Systemic Functional Grammar
STEC	Shared Task Evaluation Challenges
T2T	text-to-text
TAG	Tree-Adjoining Grammar
TSD	International Conference on Text, Speech and Dialogue
TF-ISF	Term Frequency-Inverse Sentence Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TF	Term Frequency
WebNLG	International Workshop on Natural Language Generation and the Semantic Web

Chapter 1

Introduction

As society advances, a new era of digital ecosystems is emerging, where collaborative environments are created between humans and machines. Technology is inherent to the consumer products that are now part of day-to-day human life, such as cars, mobile phones, computers and TVs. This makes it necessary for human-computer communication and interaction to be as sound, precise and as natural as possible (Jacko, 2012). This communication may imply different levels of difficulty depending on how this communication is made. In this sense, when the communication is established in the domain of the machine (e.g., using a programming language or pushing a button of an application's interface), the ambiguity is impossible due to the rules implicit in this type of communication. However, when the communication is through the use of natural language, its flexibility and ambiguity become unavoidable for a machine (e.g., when asking the virtual assistant for some information or looking it up via a search engine).

Within the Artificial Intelligence (AI) field, Natural Language Processing (NLP) is responsible for addressing the automatic analysis and representation of human language, facilitating the communication between humans and machines (Cole, 1997). This research field provides the necessary techniques that aim to comprehend and generate natural language. Within NLP, there are two main distinctions, namely Natural Language Understanding (NLU) and Natural Language Generation (NLG). The former usually addresses the search, retrieval, classification and extraction of information while the latter aims to produce and deliver the appropriate information in the most suitable manner given its established communicative purpose (e.g., to inform, summarize, report, persuade, promote, encourage or assist).

The NLP field covers a wide range of tasks, among them we can highlight machine translation (Tantuğ & Adalı, 2018), information retrieval systems (Berger & Lafferty, 2017), automatic summarization (Hardy & Vlachos, 2018) or NLG (Munigala et al., 2018). Each of them deals with the natural language in different ways, processing it automatically and taking into account several levels of language analysis, such as: (i) phonetic and phonological analysis; (ii) lexical-morphological analysis; (iii) syntactic analysis; (iv) semantic analysis; or (v) pragmatic analysis.

The importance and repercussion of the research in areas within the scope of AI and NLP has increased in recent years. In this sense initiatives are currently underway to promote these two areas. Examples of such initiatives, at the national level, can be found in the "*Estrategia Española de I+D+I en Inteligencia Artificial*[1]" (Spanish R+D+I Strategy in Artificial Intelligence) or in the "*Plan Nacional de Impulso de las Tecnologías del Lenguaje*[2]" (National Plan for the Promotion of Language Technologies). At European level, such examples can be found in "Horizon2020: research and innovation in the area of information technologies, e.g. content technologies, multilingual internet and artificial intelligence"[3] or in "Connecting Europe Facility (CEF): the e-translation solution"[4]. In addition to this, there are many applications and resources based on the NLP techniques that have been developed to deal with language in one way or another. From a broad perspective, there are some applications that people use daily without knowing that they may employ NLP under their surface. For example, when we look up for something in a search engine (e.g., Google[5] or DuckDuckGo[6]), when you travel to another country and you need to automatically translate text using an application (e.g., DeepL[7] or Google Translate[8]) or when you ask a virtual assistant for information (e.g., Siri[9], Alexa[10] or Google Home[11]). From a professional perspective or in the research field, NLP techniques or applications are used in many ways. For example, there are applications that work at word level that are useful to obtain information related to a specific word. This is the case of lexicons such as WordNet (Fellbaum, 1998) that stores information about the synonyms of a word as well as other lexical-semantic relationships between words; lemmatizers that extract the lemma of a given word; or Part-Of-Speech (POS) taggers which provide information about the lexical categories of the words (e.g., noun, adjective, adverb, verb, etc.). As in the case of words, some applications also work with sentences and full documents. For instance, language analyzer tools as Freeling (Padró & Stanilovsky, 2012) or the Stanford CoreNLP tool (Manning et

[1] http://www.ciencia.gob.es/stfls/MICINN/Ciencia/Ficheros/Estrategia_Inteligencia_Artificial_IDI.pdf

[2] https://www.red.es/redes/es/que-hacemos/tecnolog%C3%ADas-del-lenguaje

[3] https://ec.europa.eu/programmes/horizon2020/en/h2020-section/information-and-communication-technologies

[4] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eTranslation

[5] https://www.google.com

[6] https://duckduckgo.com/

[7] https://www.deepl.com/translator

[8] https://translate.google.com/

[9] https://www.apple.com/es/siri/

[10] https://developer.amazon.com/es/alexa

[11] https://store.google.com/es/product/google_home

al., 2014) can be used to analyze a text at different levels (i.e., syntactic, lexical and semantic levels).

Up to now most of the research effort in NLP has been focused on NLU, relegating the NLG task to the mere extraction of literal text fragments (Vodolazova et al., 2013), copy-and-paste techniques (Jing & McKeown, 2000), use of templates (Mitchell et al., 2014) and domain-specific static approaches which produce language by means of restricted vocabularies and grammars (Bouayad-Agha et al., 2012; Androutsopoulos et al., 2013). Therefore, in this thesis, we focus on NLG. The remainder of this chapter will describe this task in more detail and the motivation behind this research (Section 1.1). The main objectives of this work are detailed in Section 1.2. Then, the research projects related to this thesis are outlined in Section 1.3. Finally, the structure of the thesis is delineated in Section 1.4.

1.1 Motivation and Problem Definition

Among the disciplines included in the NLP field, NLG is the one in charge of automatically developing techniques to produce human utterances, in the form of text or speech (Bateman & Zoch, 2003). The NLG is a multidisciplinary task. Its research and development include knowledge from diverse areas such as linguistics, psychology, engineering and computer science. As mentioned before, the main objective of this discipline is to investigate how to create computer applications capable of automatically producing high quality texts in natural language. For this purpose, the NLG process starts either from structured and processable data representations (binary files, numerical data, databases, etc.) or from texts written in natural language. The transformation of these data cannot be carried out directly. In fact, many decisions have to be made involving different aspects, including, for example: the determination of the message content and structure; rhetorical relationships at various levels (text, paragraph, sentence); the choice of appropriate words; the final layout of the text (title, headers, footnotes, etc.); and, acoustic patterns in the event that the final output of the message is oral. One of the greatest challenges of NLG is the construction of architectures in which all these decisions can be made in such a way that the production of texts is temporarily affordable, whether in text or audio format (Bateman & Zoch, 2003).

There are very broad and varied applications of NLG. Systems exist on the market that are responsible, for example, for the generation of weather reports (Ramos-Soto et al., 2015). There are also systems that generate simplified texts (Macdonald & Siddharthan, 2016), academic evaluation reports (Williams & Reiter, 2008) or that create summaries (Hardy & Vlachos, 2018). These are just a few applications, but they will be reviewed in more detail in Chapter 2.

Traditionally and due to the complexity of the NLG task, NLG systems have used a hybrid approach, combining different components using different techniques (Bangalore & Rambow, 2000). However, one of the limitations of NLG systems is that they have been designed for very specific domains (Ramos-Soto et al., 2015) and for a specific purpose (Ge et al., 2015), with the development of open and flexible domain approaches still a challenge for the research community (Barros & Lloret, 2017). In addition, there is currently a new added challenge to this task. This challenge is related to the large number of heterogeneous information sources belonging to different textual genres (e.g., such as news, technical documentation, regulations, blogs, reviews, forums, social networks, etc.). Therefore, it is indispensable to study these sources in order to understand the characteristics of each of them to be able to design methods that are independent of the data sources, the domain and the textual genre to which they belong.

Based on the limitations of the existing NLG systems, where their design prevents their adaptation to other domains and purposes, the starting hypothesis of this thesis is that the application of a hybrid approach for the generation of natural language will increase the quality of the language produced, favouring its independence of domain, the textual genre and the final application that uses it.

1.2 Objective of the Thesis

Based on the hypothesis defined in section 1.1, a main objective and several sub-objectives are proposed. The main objective of this thesis is to analyze, research and propose a novel hybrid approach for NLG, combining statistical techniques with knowledge-based ones. The proposed approach needs to be flexible enough and easily adaptable to different domains and scenarios. In order to successfully achieve this goal, the following sub-objectives are defined:

1. To carry out an exhaustive state-of-the-art review in the field of NLG, analyzing the existing approaches, evaluation methodologies and existing resources at present.

2. To investigate, propose and analyze new approaches for NLG using techniques based on NLP, focusing mainly on hybrid language generation approaches that combine statistical and knowledge-based methods.

3. Thoroughly evaluate the proposed approach, using standard metrics or adapting the evaluation frameworks to the characteristics of the scenarios or domains. The evaluation will consist of both intrinsic and extrinsic as well as quantitative and qualitative evaluations and will be compared, as far as possible, to existing approaches.

4. Consider the application of the proposed approach in the context of other NLP tasks, in particular, text summarization.

5. Conclusions and benefits of this research together with a proposal for on-going and future work.

1.3 Context: Research Projects Related to the Thesis

The research work presented in this thesis is related to some of the activities that were developed in three different research projects whose execution was carried out in parallel to the development of this thesis. The research work carried out for this thesis has taken place within the framework of the different research projects that were developed by the Language Processing and Information Systems (GPLSI) research group of the Department of Software and Computing systems (DLSI) of the University of Alicante. In this way, the thesis benefited from the ideas developed in these projects, and conversely, the projects benefited from the findings and knowledge-building that has resulted from this thesis, thus creating a perfect symbiosis. A brief description of each GPLSI research project is provided next, together with their respective objectives, underscoring the aspects that are related to this thesis.

- **DIIM2.0: Development of intelligent and interactive techniques for information mining and generation on the Web 2.0** [12] **(PROMETEOII/2014/001)**, Duration 60 months
  The main objective of this project was to work on the resources, tools and systems of text mining providing them with the necessary functionality to favor their transference to society. For this purpose, new ways of applying text mining and opinions to business processes, their connection to Business Intelligence systems, and their capability as tools to help decision making were investigated.
  In order to achieve this objective the following lines of research were proposed:

  - To research and to develop new techniques for the construction of specialized systems for mining texts and opinions. Research was focused on the following areas:
    - To develop, search, retrieve and classify systems for existing information in web 2.0 and large documentary collections for the extraction of relevant information. Specifically, the adaptability of the tools to the different user profiles was investigated in order to obtain the information grouped as a ranking of results in accordance with the specific needs of the business. Special work was done on the incorporation of automatic learning techniques to obtain user profiles and their applicability to process, retrieve and sort the search.
    - To develop information extraction tools for business intelligence systems, from publications in web pages, press media, official

[12] https://gplsi.dlsi.ua.es/gplsi13/es/node/383

bulletins, and other digital media, that are useful for the conducting of market studies, competition reports, etc., all of which improve the competitiveness of organizations.

    - To develop sentiment analysis tools capable of mining social networks in order to automatically extract ratings on brands, products, people and issues in general; from the opinions expressed by users, and then perform an analysis of behaviors and trends by managing comparisons for time intervals or geographical areas.

  - To research and develop new techniques for the construction of human language generation systems. Research was focused on the following areas:

    - To develop automatic text reconstruction systems, in order to produce new simplified or enriched documents to assist people with some kind of reading disability, or to adapt texts to a specific target audience (adaptations for children, inexperienced users, etc...). Special attention was given to defining a model for understanding the source, integrating resources and making data compatible in order to automate the reconstruction process.
    - To develop automatic production systems for summaries and reports, to facilitate decision-making in the business environment and in general, in any type of organization. Research was focused on obtaining relevant information and its integration into reports and business intelligence systems.

This thesis was directly related to the second line of research previously described in the project objectives (construction of human language generation systems). Specifically, the research conducted in thesis is related to the activities "Research into automatic text reconstruction techniques" and "Research into automatic production techniques for summaries and reports". The work of this thesis has contributed to the project with the creation of a novel NLG approach (which will be described in Chapter 4) and whose performance will be demonstrated in Chapters 5 and 6.

- **LEGOLANG: Deconstruction Techniques applied to Human Language Technologies** [13] **(TIN2012-31224)**, Duration: 36 months
  The main objective of this project was based on the need to rethink the classical philosophy of NLP in order to adapt both to the sources available (unstructured data with multi-modality, multi-language and different degrees of formality) and to the real needs of end users. In order to achieve this objective, it was necessary to integrate both the understanding and the generation of human language in a single model (LEGOLang model) based

[13] http://gplsi.dlsi.ua.es/legolang/

on language deconstruction techniques, independent of its final application and of the variant of human language chosen to express knowledge. In order to achieve this objective, the following sub objectives were proposed:

  - To define and develop a multidimensional structure of knowledge storage that will act as an object of exchange between the processes of comprehension and generation of human language, and that will allow the creation of a repository of knowledge for its later use.
  - To create, compile and adapt NLP resources, techniques and tools for understanding and generation of human language and its integration into the LEGOLANG structural model.
  - To create an evaluation framework for the model based on the combination and unification of intrinsic metrics to the components of the model, as well as the development of a use scenario that shows its validity.
  - To promote and disseminate the research lines of the project through the participation and organization of activities in campaigns, conferences, workshops, seminars and thematic networks, as well as the possible technological transfer to society.

The research conducted in this thesis is related to the activity "*GLH.REAL: Realización*" (Realization), of the NLG module of this project. The main objective of this activity was to transform the information contained in elements called *L-Bricks* into human language. However, the beginning of this thesis work coincided with the end of the project. Therefore, this thesis provided information for the review of state of the art in NLG, specifically for the task of surface realization.

- **RESCATA: Canonical Representation and transformations of texts applied to the Human Language Technologies** [14] **(TIN2015- 65100-R)**, Duration: 36 months
  The main objective of this project was based on the need to investigate a new paradigm for text comprehension that allowed us to determine a standard, unique, invariable and independent representation. This representation is called canonical representation, from which, different types of inflections are obtained through a transformation process. These inflections are appropriate to the needs of each user to be applied to other NLP tasks, such as simplifying, enriching or summarizing.
  In order to achieve this objective, the following sub objectives were defined:

  - To define what is the canonical representation of texts, identifying the information necessary to obtain a representation and developing a

[14] http://gplsi.dlsi.ua.es/rescata/

service structure that guarantees that this representation is a standard, unique and invariable form of the knowledge contained in those texts.

- To define the text inflections, identifying the information necessary for the generation of such inflections and developing a service structure that allows such variations to be obtained.
- To create, compile, and adapt NLP resources, techniques, and tools for understanding and integrating them into the canonical structural model.
- To identify the needs of users and their relationship with the different inflections that can be generated from the canonical representation of texts.
- To create an evaluation framework for the model, based on the combination and unification of metrics intrinsic to the components of the model, as well as develop a use scenario that shows the validity of the model.
- To promote and disseminate the lines of research of the project through the participation and organization of activities in campaigns, conferences, workshops, seminars and thematic networks, as well as the possible transfer of technology to society.

In particular, the research work carried out in this thesis is related to the module B: Inflections. The objective of this module was to define the concept of inflection, the possible types of inflections that a canonical form may have, and finally to determine what information will be necessary, taking into account the different levels of linguistic analysis for the NLP. More specifically, the work of this thesis is directly related to the activities B.1 Definition of the inflections and B.2 Definition of the inflection type. In this respect, this thesis provides the adaptation of our approach to summarization tasks, where these inflections can be used.

1.4 Thesis Structure

This thesis is structured around 6 chapters that meet the objective and sub-objectives described in Section 1.2. In this regard, Chapters 2 and 3 are related to the first sub-objective; Chapter 4 is related to the third one; Chapters 5 and 6 are related to the fourth and fifth sub-objectives respectively; and, finally, Chapter 7 is related to the last sub-objective. Each of these chapters explores some of the issues related to the Natural Language Generation (NLG) research area.

- **Chapter 2: Background in Natural Language Generation.** This chapter provides an overview of the state of the art in NLG, ranging from a description of the different architectures employed in this research area to an analysis of the methods using them for each of the NLG stages.

- **Chapter 3: Resources and Evaluation in Natural Language Generation.** This chapter provides an overview of the existing resources for the NLG field, including tools and corpora. In addition to this, the evaluation methodology for this area is outlined as well as information about the relevant NLG conferences and workshops.

- **Chapter 4: HanaNLG: A Flexible Hybrid Approach for Natural Language Generation.** This chapter describes the methodology followed for tackling the NLG from a hybrid perspective. In this sense, HanaNLG, a hybrid surface realization focused approach is explained along with the modules involved in it.

- **Chapter 5: HanaNLG Intrinsic Evaluation.** This chapter contains the evaluation environment in which HanaNLG was evaluated. The chapter comprises the experiments carried out together with its results. In this regard, an intrinsic incremental evaluation was performed, where the results obtained in one experiment will justify the decisions made for the following experiments. This will allow the evaluation of every aspect concerning HanaNLG

- **Chapter 6: HanaNLG Extrinsic Evaluation: Application in Automatic Summarization Tasks.** In this chapter, an extrinsic evaluation of HanaNLG is carried out not only focusing on the generated text but how well the goal of a specific application is met. In this sense, we analyzed the performance and adaptability of HanaNLG to two applications of the text summarization field: headline generation and cross-document timeline generation.

- **Chapter 7: Conclusion and Work in Progress.** This chapter summarizes the main conclusion of this research work and the main contributions of this thesis. It also addresses some issues that will be faced in the future. Finally, a list of the relevant publications related to this thesis is provided.

- **Appendix A: Resumen.** Provides a digest of the thesis in Spanish. This summary contains the objectives, the main contributions and findings of the thesis, as well as explaining the most relevant experiments carried out and the results obtained.

CHAPTER 2

Background in Natural Language Generation

The task of automatically generating natural language NLG has developed extensively since its inception at the beginning the 1950s. This has given rise to the creation and development of new techniques and paradigms to address this field. Some of these works have been mentioned in recent NLG state of the art surveys (Gatt & Krahmer, 2018; Ramos-Soto et al., 2016; M. E. Vicente et al., 2015) emphasizing the different ways of approaching NLG as well as the difficulties involved.

Since NLG is a challenging task, the main objective of this chapter is to present a review of its state of the art. This review is important to the present thesis for two reasons: (i) knowing how the current NLG systems are developed will provide insights on what is needed to advance the field; and (ii) deciding the types of techniques to employ along with the architecture to develop a novel approach is only possible after reviewing the state of the art. In this regard, a description of the existing types of systems and techniques in addition to the architectures widely used is provided.

The rest of the chapter summarizes the main background of the NLG area, including an exhaustive analysis of the techniques employed in this research field. To properly introduce the NLG area, Section 2.1 provides an analysis of the different ways that a NLG system can be classified, either by the type of input to the system (Subsection 2.1.1) or by the purpose for which the system was created (Subsection 2.1.2). Section 2.2 provides the characteristics of the most used architecture during the development of a NLG system. Then, Section 2.3 describes the approaches that have been used to address the NLG problem. In particular, NLG is commonly addressed with two distinct approaches. While knowledge-based approaches rely on linguistic information to produce text (Subsection 2.3.1), statistic ones generate text based on probabilities extracted from a volume of documents (Subsection 2.3.2). In addition to these approaches, the use of hybrid approaches, which combine the knowledge-based and statistic perspectives (Subsection 2.3.3), and deep learning (Subsection 2.3.4), are gaining popularity in the NLG research field. Finally, Section 2.4 concludes the Chapter by providing some insights on the future directions of NLG.

2.1 Classification of NLG Systems

In this classification we mainly focus on the generation of text, so speech generation will not be considered. A NLG system can be classified according to different criteria. The study undertaken in this thesis exposes that there are two indispensable factors to take into account when addressing the generation task: i) the input to the system and ii) the target of the system. These factors were selected as the criteria for classifying the generation systems due to its importance, since there are different types of inputs and the systems may be created for diverse purposes. Table 2.1 shows a summary of the different types of classification for a NLG system.

Classification	Type	Definition
System input	*D2T*	Receive as input numeric data, databases, knowledge bases or tagged corpus
	T2T	Receive as input text or sentences
System target	Informative Texts	Generate reports
	Summaries	Generate a brief synthesis of the input
	Drafts	Generate draft versions of technical texts
	Simplified texts	Simplify the input text to make it easier to understand
	Persuasive texts	Generate text with the aim of convincing or motivating users about a topic
	Dialogue systems	Generate interactive texts maintaining a communication between the user and the system or between user and chatbots
	Reasoning explanations	Expose in a text the steps followed in the resolution of a problem
	Recommendations	Generate suggestions and assessments regarding places, products, services, etc

Table 2.1: Classification of NLG systems

The type of systems summarized in the table will be detailed below.

2.1.1 Classification by the input of the system

According to the type of input introduced to the system, two strategies are considered in NLG: data-to-text (D2T) and text-to-text (T2T). While in the D2T perspective the input to the system is a set of data which do not form a text itself (e.g. numeric data representing information of the vital signs of a patient given by a sensor); in a T2T system a text is taken as the input whose relevant information is used to construct the final output of the system.

Data-to-Text

The input format for D2T systems can change from one system to another, with numeric data commonly used as input to the system, despite not being the only option for these types of systems. Therefore, systems that use this type of information as input are commonly found (e.g. information coming from sensors, weather stations, medical devices, etc). But it is also common to consider other structured data sources such as tagged corpus, databases, knowledge bases, log files, etc. Some authors employ the term "*concept*" to refer to this type of non-linguistic data, which is why this approach is also mentioned as concept-to-text (Barzilay & Lapata, 2005; Konstas & Lapata, 2013).

Proteus (Davey, 1974) is an example of this type of system. This system generates a summary of a tic-tac-toe[1] from a list of movements. The following is a list of movements constituting a game between *Proteus* and the author, Davey:

```
P:1 D:3 P:4 D:7 P:5 D:6 P:9
```

Whose output, constructed from the point of view of the system is:

> *The game started with me taking a corner, and you took an adjacent one. I threatened you by taking the middle of the edge opposite that and adjacent to the one which I had just taken but you blocked it and threatened me. I blocked your diagonal and forked you. If you had blocked mine, you would have forked me, but you took the middle of the edge opposite of the corner which I took first and the one which you had just taken and so I won by completing my diagonal.*

Another example of this type of system is PASS (Van der Lee et al., 2017), a D2T system that generates soccer reports. This system creates a summary of a specific match employing a template-based approach. The input to this system is match statistics and heterogeneous data such as the league, the date, the match events, the players, the total number of shots, the possession ball percentage or the accuracy of the passes. In (Anselma & Mazzei, 2018), a system that automatically generates messages for diet management is proposed. This system performs

[1] Tic-tac-to is a simple game for two players, who take turns to place X or O in the spaces of a 3x3 grid. This game is also known as noughts and crosses or Xs and Os.

numerical computation combining several factors related to diet requirements and energetic information about the food and reports the results by using NLG.

Apart from these D2T systems, there is a new trend which aims to understand and describe the actions associated to diagrams or models of business process. Examples of this type of system can be found in (Delicado et al., 2017), where several NLP tools for business process management (including one for generating text) are presented; and in (Aysolmaz et al., 2018), where a semi-automatic process to analyze business models and generate a requirement document describing these models is proposed. The former generates the sentences using deep-syntactic trees while the latter generates text with the use of templates.

Text-to-Text

The systems addressing the NLG with this type of approach, can take as input both, texts or individual sentences. There are many applications in the NLG task which employ T2T systems such as summarization, answer merging, text simplification, text compression or headline generation.

An example of this type of system can be found in (Sauper & Barzilay, 2009), taking as input a set of documents from the Internet. Wikipedia articles are generated whose structure is determined by the domain to which the production belongs (e.g. diseases articles include four sections: diagnosis, causes, symptoms and treatment). In (Li et al., 2018) an abstractive summarization approach is proposed. In this approach, taking as input a document, a set of salient sentences are first selected and then are compressed and paraphrased to create a summary, which is known as an abstractive summary. Angrosh and Siddharthan (2014) presented a text simplification method based on synchronous dependency grammars which takes as input a sentence. This method combines lexical rules with hand-crafted syntactic rules to create a simplified version of the input sentence. An example of sentence compression can be found in (Filippova et al., 2015), where a LSTM approach to deletion-based sentence compression is described. In this approach the sentence is translated into a sequence of zeros and ones which corresponds to the characters that need to be removed. In (Colmenares, Litvak, Mantrach, & Silvestri, 2015) an approach for headline generation is proposed. In this approach, the headline generation task is addressed as a discrete optimization task in a feature-rich space, using a conditional random fields model with state transitions. This model tries to learn how humans construct a title for a news article by means of a mapping function that transforms headlines into a abstract feature rich space where the characteristics to discerning that a headline have been produce by a human can be identified.

2.1.2 Classification by the target of the system

As stated above, NLG systems can also be classified depending on the target for which the system was created. Depending on the type of application, the target of the system may differ. For example, the system target is not the same for a system that generates summaries as for a system that generates persuasive texts. The most relevant system targets have been collected in this study:

- *Informative texts generation.* The purpose of this system in this case is to generate reports from factual data (objective information). *FoG* (Goldberg et al., 1994) and *SumTime* (Reiter et al., 2005a) are two systems of this type. These systems take as input numeric information from simulation systems which represent magnitudes as the temperature, rainfall level or the speed of the wind in different places and at different times. There are applications in other contexts, such as *SkillSum* (Williams & Reiter, 2008), which is a tool that generates reports on academic assessments. The aim of this tool was to help people with little knowledge of arithmetic and language. Other examples can be found in (Gkatzia et al., 2016) where medical reports are generated from time-series data; or in (Nesterenko, 2016) where financial reports are generated from stock news. Example 1 shows a report generated by *SkillSum*.

  (1) **English skills**
  Thank you for doing this.
  You got 17 questions right. Click here for more information.
  Your skills may not be OK for your construction course.
  It looks as if you find punctuation quite hard.
  You got all except 2 of the reading questions right. But you made 8 mistakes on the questions about writing.
  Perhaps you would like to take a course to help you with your
  punctuation.
  An English course might help you, because you said you do not feel that your reading is very good.
  Click here for Key Skills at Xshire College.

- *Summarization.* This type of language generation aims to produce an abridged version of one or more information sources. These summaries can be associated with diverse fields: medical summaries (Hunter et al., 2012), engineering summaries (Yu et al., 2007), news summaries (Hardy & Vlachos, 2018) or patent summaries (Mille & Wanner, 2008), among others. Example 2 shows a snippet of a generated medical summary extracted from (Hunter et al., 2012).

  (2) ...
  Respiratory Support
  Current Status
  Currently, the baby is on CMV in 27 % O2. Vent RR is 55 breaths per minute. Pressures are20/4 cms H2O. Tidal volume is 1.5.
  SaO2 is variable within the acceptable range and there have been some desaturations.
  The most recent blood gas was taken at around 07:45. Parameters are acceptable. pH is 7.3.CO2 is 5.72 kPa. BE is -4.6 mmol/L. The last ET suction was done at about 05:15.

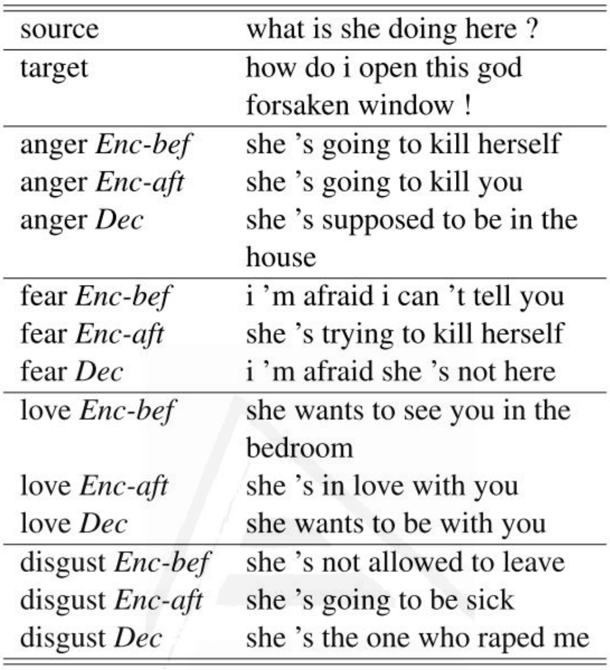
> **Events During the Shift**
>
> A blood gas was taken at around 19:45. Parameters were acceptable. pH was 7.18. CO2 was7.71 kPa. BE was -4.8 mmol/L.
>
> Another ABG was taken at around 23:00. Blood gas parameters had deteriorated to respiratoryacidosis by around 23:00. pH was 7.18. CO2 had risen to 9.27 kPa by around 23:00. BE was -4.8 mmol/L.
>
> The baby was intubated at 00:15 and was on CMV. Vent RR was 50 breaths per minute.Pressures were 20/4 cms H2O. FiO2 was 29 %. Tidal volume was 1.5. He was given morphineand suxamethonium. MAP was raised from 6 cms H2O to 8 cms H2O.
>
> Between 00:30 and 03:15, SaO2 increased from 88 % to 97 %
>
> ...

- *Generation of simplified texts.* These systems have been designed as tools to help people with disabilities or with reading comprehension problems, regardless of whether the cause is unfamiliarity with the language or cognitive difficulty. There are systems of this type that produce text aimed at aphasic people (Reiter et al., 2009) or text that allows blind people to examine graphics (Ferres et al., 2006). *SkillSum*, previously mentioned as an example of report generation, incorporated techniques for producing texts that can be read by quasi illiterate people. There are also systems to make complex texts more accessible for low-literacy readers (Siddharthan, 2014) or for children (Macdonald & Siddharthan, 2016). An example of a simplified news story for children, extracted from (Macdonald & Siddharthan, 2016), is shown in Example 3

(3) A new species of titanosaur unearthed in Argentina is the largest animal ever to walk the Earth, palaeontologists say. Based on its huge thigh bones, it was 40m (130ft) long and 20m (65ft) tall. A film crew from the BBC Natural History Unit was there to capture the moment the scientists realised exactly how big their discovery was. This giant herbivore lived in the forests of Patagonia between 95 and 100 million years ago, based on the age of the rocks in which its bones were found. There have been many previous contenders for the title “world’s biggest dinosaur."

- *Generation of persuasive texts.* In (de Rosis & Grasso, 2000), the term *affective* is used to characterize those NLG systems that seek to influence or take into account the emotional state of the listener. The language can be employed to achieve multiple purposes (e.g. motivate, persuade, reduce stress, etc.). Among the systems designed under these premises, we can find, as an example, *STOP* (Reiter et al., 2003), a generator of letters that aims to dissuade users from smoking; or a system that seeks to reduce the anxiety of cancer patients by providing them with information (Cawsey et al., 2000). Other systems can provide entertainment, either through riddles (Binstead & Ritchie, 1997) or stories (Laclaustra et al., 2014) or provide persuasive texts about fashion products (Munigala et al., 2018). An example of a persuasive text about fashion products, extracted from (Munigala et al., 2018), is shown in Example 4.

(4) Look sexy in your beachside look with mid lx navy sneakers using a cool hip-hop look



source	what is she doing here ?
target	how do i open this god forsaken window !
anger *Enc-bef*	she 's going to kill herself
anger *Enc-aft*	she 's going to kill you
anger *Dec*	she 's supposed to be in the house
fear *Enc-bef*	i 'm afraid i can 't tell you
fear *Enc-aft*	she 's trying to kill herself
fear *Dec*	i 'm afraid she 's not here
love *Enc-bef*	she wants to see you in the bedroom
love *Enc-aft*	she 's in love with you
love *Dec*	she wants to be with you
disgust *Enc-bef*	she 's not allowed to leave
disgust *Enc-aft*	she 's going to be sick
disgust *Dec*	she 's the one who raped me

Figure 2.1: Example of responses generated by different models with emotions anger, fear, love and disgust.

- *Dialogue systems.* Dialogue systems are oriented towards the communication between people and machines. Their main characteristic is that the user interacts with the system, which generates sentences in natural language conditioned by the immediately previous context. Systems of this type have been designed for multiple purposes, such as helping to improve writing skills (M. Liu et al., 2012), or tutoring to increase the knowledge of certain subjects through dialogue (Dzikovska et al., 2011). There are other systems which generate dialogues that express emotions (Huang et al., 2018) or others that are used in virtual game environments (Koller et al., 2009). Within this type of system, some applications have been developed for chatbots (Xing & Fernández, 2018). Figure 2.1 shows an example of a dialogue with emotions, extracted from (Huang et al., 2018).

- *Generation of reasoning explanations.* The output of this type of system is the explanation of a sequence of steps that the system followed in the execution of an algorithm, the processing of a transaction, the resolution of a mathematical problem, etc. An example of these systems is *P.Rex* (Fiedler, 2005), a tool for explaining theorem demonstrations; or a system that generates explanations for a machine learning algorithm decision (Forrest et al., 2018; Alonso et al., 2018). The approaches for business process management previously mentioned in Section 2.1.1 (i.e. (Delicado et al., 2017) and (Aysolmaz et al., 2018)) are also examples of these systems. This is a new research field within NLG; hence, there are not many examples of them, but it is starting to gain importance. Example 5 shows an explanation generated which has been extracted from (Forrest et al., 2018).

(5) An automated explanation tool has been used to create the explanation below. It shows the influence each variable had on the decision to give or refuse credit.
The decision reached by the algorithm is to refuse credit. The explanation tool has examined the values of the input variables. Their total influence on the algorithm was 81.0% to refuse credit, versus 19.0% to give credit.
The single greatest contribution to the decision is from the variable 'current account' with the value of 'in debit' this produced 40% of the whole decision, influencing the algorithm to refuse credit. Other important variables were 'assets' with the value 'none' and 'savings account' with the value 'less than 100', these influenced a decision to refuse credit. Minor influences on the algorithm to refuse credit were 'duration' with the value '24' and 'credit value' with the value '4870'.
Minor influences on the algorithm to give credit were 'housing' with the value 'free', 'credit history' with the value 'delayed payment', 'other credit' with the value 'none' and 'purpose' with the value 'car (new)'.

- *Generation of recommendations.* Nowadays, there is an increasing number of sites where users can share information about their tastes and opinions. As a consequence, tools are being developed to process and translate these data into recommendations and trends. In this context, it makes sense to develop systems that refer to any field or subject, such as restaurants, cinemas, tourist destinations or technological products, providing results of processing in natural language. A system that does this is *Shed* (Lim-Cheng et al., 2014), which, based on the user profile and obtaining information from Web 2.0 (reviews, tweets, etc.), recommends personalized nutritional diets. Another example can be found in (Conde-Clemente et al., 2018), where an adapted recommendation on more responsible consumption is generated for households with inefficient pattern of energy consumption. An example of a energy consumption recommendation report, extracted from (Conde-Clemente et al., 2018), is shown in Example 6.

(6) Household ID: 144283
Analyzed period: from 2014/01/01 to 2014/02/28
Physical cluster: Flats with maximum 2 adults with some systems, e.g., boiler and airconditioning.

Dear householder, we know that you are not very concerned with energy consumption but we would like to provide you a set of tips to improve the sustainability of the planet.

General consumption: Your average consumption is more than double with respect to households similar to you. If you reduce it, you will improve your energy efficiency.

> Do it for the planet!
>
> *Specific consumption:* You are consuming the most part of the energy during those periods in which the energy cost is higher. If you want to save money on your bill, you should shift part of your electrical consumption from the morning (high cost) to the dawning (low cost).
>
> *Standby consumption:* Your standby consumption is higher with respect to households similar to you. If you reduce it, you may save money on your bill. Yes, you can!

In addition to these types of systems, there are others that belong to the computational narratology and computational creativity fields, which are two emerging areas of the AI. Several works that connect these two areas with the NLG have been developed. In this regard, the work presented in (Gervás, 2017) shows an approach to create a poem with thematic consistency and enjambment [2] though the use of an n-gram based method (see Section 2.3.2). In (Manjavacas et al., 2017) a co-creative text generation system for producing science fiction literature using character-based Language Models (LM) is proposed. Other example can be found in (Alnajjar & Hämäläinen, 2018) where a system that automatically generates satirical movie titles is proposed; or in (M. E. Vicente et al., 2018), where an automatic story generation approach is presented.

2.2 Architecture of NLG Systems

The previous section presented how NLG systems can be classified, but their processes may differ from one to another. From a general perspective, the NLG process could be described as the accomplishment of a set of tasks whose purpose is to transmit, in natural language, certain information to an audience in order to achieve an objective. Therefore, for the system, characterizing the input and the output is just as important as considering the context and the communicative goal. Specifying the tasks or stages so that the system fulfills its purpose implies that each of them must contemplate and concretize such aspects. In this regard, many architectures have been proposed (Kantrowitz & Bates, 1992; Hovy, 1988; Calder et al., 1999; García Ibáñez et al., 2004; Mellish et al., 2006), but the most employed so far in the NLG community is the one presented by Ehud Reiter and Robert Dale (Reiter & Dale, 2000). According to their architecture, the functionalities that correspond to a NLG system, which will be described below, are distributed among seven tasks and the relationship established between them can be represented through a basic architecture of three stages:

- *Macroplanning.* The first stage of the system must determine what to say and organize it into a coherent structure, resulting in a document plan.

[2] In poetry, a metric phenomenon that occurs when there is a disagreement between a syntactic unit and a metric unit.

  This is done through two tasks:

  - Content selection
  - Structuring the document

- *Microplanning.* Starting from the document plan from the first stage, a discourse planning will be generated. The appropriate words and references are selected, the messages are provided with a linguistic structure and the information is grouped into sentences. The tasks involved are as follows:

  - Sentence aggregation
  - Lexicalization of syntactic structures
  - Generation of referring expression

- *Realization.* At this point of the process, there is a representation of the sentences that will comprise the output of the system. The realization stage generates the final output, whether text or speech, the specific sentences contained in it, as well as their structure. The two remaining tasks are as follows:

  - Linguistic realization
  - Realization of the structure

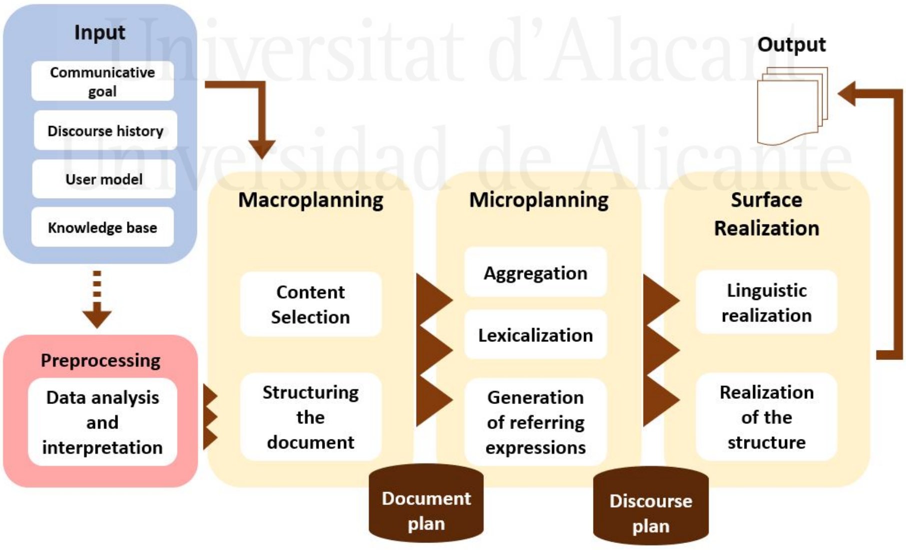


Figure 2.2: Reference architecture for a general NLG system

Figure 2.2 shows this distribution of functionalities in three main blocks. Sometimes, a data preprocessing block is added, but this will be discussed in more detail in the next section (Section 2.2.1). This type of architecture is considered to be a sequential architecture. In this type of architecture, the transformation processes of information take place unidirectionally, which prevents revisions or the possibility of modifying what has been established in past stages.

2.2.1 Macroplanning

The macroplanning stage is also known as the document planning stage. It involves making decisions as to what information to include in the system output and determining the structure it will adopt (how to organize this information).

In Section 2.1.1 the NLG systems were classified according to the type of input. In this sense, a difference was established between those systems in which the input was a text and those whose input was composed of a set of structured data. Delving into this initial distinction, what is used as input can take many forms, such as daily forecast records, sensor information or user queries.

In this way, as reflected in Figure 2.2, some authors add a pre-processing stage prior to the macroplanning stage (Reiter, 2007). This stage would be necessary when the input data have to be analyzed and interpreted. The analysis is responsible for extracting patterns from the data while the interpretation stage is performed to infer messages in the domain of the application.

The output of this module is called the document plan, which usually takes the form of a tree with messages in its terminal nodes. Messages are elementary units of discourse from the domain that can be expressed through sentences. And next to the messages, information related to the way they relate is incorporated. Figure 2.3 is the document plan associated to Table 2.2. It belongs to the aforementioned system SumTime (Reiter et al., 2005a) (see Section 2.1.2) that processes meteorological information.

Hour	Direction	Wind speed
06:00	SE	11
09:00	SSE	13
12:00	SSE	14
15:00	SSE	15
18:00	SE	18
21:00	SE	23
00:00	SE	28

Table 2.2: Input for SumTime. Wind Forecast for September 19, 2000

As can be seen in Figure 2.3, the root node indicates that the information should be contained in a paragraph. It also provides information about the order between nodes and the value of the parameters they share.

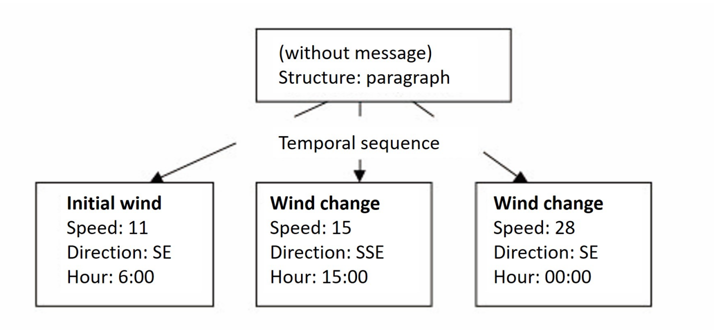


Figure 2.3: SumTime document plan

Content Selection

Content selection is the task that allows the system to choose and obtain the information that should be communicated in the final text: the most relevant information for the user according to the communicative goal and the situation, which includes aspects as diverse as the size that corresponds to the output of the system, the level of knowledge of the user or what has been generated so far.

Since this stage is the least related to linguistic processing, some authors have placed it outside the NLG system. This is the case of (Evans et al., 2002), where the authors propose a new frontier that excludes from the discipline those actions whose nature is not strictly linguistic. A more flexible approach is proposed in (McDonald, 2010), where the author establishes a division between two applications: one 'generator', which would be in charge of the linguistic processing, and another, which he calls the 'speaker', whose function would be to determine what to say by passing this information to the 'generator'. Even with that separation, they would be considered to be part of the same system which needs them both.

Structuring the document

In order to achieve a coherent text, the elements that make it up must be properly structured. Cohesion and coherence are the principles that allow a set of sentences to constitute a discourse. They refer to the way in which textual units relate to each other and permit inferences to be made from the information provided or the unambiguous identification of coreferential elements. This coherence concerns both the sentences and the messages that compose them.

For all these reasons, it is necessary to have a task that, either during the process of selecting the messages or after having done so, determines the structure that the final text will have, the relationship that some elements have with others, given that such arrangement is the first step towards a correct discourse.

Likewise, as in the case of the other stages in NLG, both to select the techniques to address this stage and to determine the type of structure needed, extralinguistic aspects have to be considered. A text that *explains* a process will not have the same structure as a text that *compares* two proposals, in accordance with the communicative goal. If the context is considered, a continuity with the preceding structures must be shown so as not to disorientate the user.

2.2.2 Microplanning

In the microplanning stage, the document plan is taken as the input, which is the result of the macroplanning in which the messages that must form part of the final text are indicated as well as their structure. The operations that are carried out from this document plan, in the different tasks that fall within the scope of this microplanning stage, are eminently linguistic. In order to achieve their purpose, the operations can take knowledge bases or ontologies as a source and, in order to make their choices properly, they consider both the communicative goal and the user model, that is, the characterization of the receiver.

The output of this stage is the specification of the text or the discourse plan. The text to be generated must be fully characterized in such a specification. The output takes the form of a tree in which the delimitations of the sentences, that is, their syntactic relations, the words they contain or the coreferential relations are presented.

The decisions to be made in this stage determine the tasks to follow:

- *Aggregation*, or how to group the structures from the document plan together to form coherent sets of messages.
- *Lexicalization of syntactic structures*, because the words that will be used to express the concepts and facts contained in the document plan must be determined.
- *Generation of Referring Expressions* (GRE), given that the same concept or entity may appear on different times throughout the text, the way that it is referenced or described in each appearance must be chosen.

Aggregation

In the aggregation task, the combinations to be made on the information elements included in the document plan must be determined. This task is also responsible for establishing an order between the results of those combinations. For some authors, the aim would be to remove the redundancy (Dalianis, 1996), while for others it is the combination of the messages (Cheng et al., 1997). In any case, whatever the perspective adopted, the result will deliver conciseness and syntactic simplicity to produce a coherent text (Bernardos, 2007).

The aggregation of two sentences can be carried out following different mechanisms. Some simple examples are shown below:

- *Simple conjunction*: the lexical or syntactic content of the components does not change:

"Ana is from Murcia. Luis is from Albacete."
"Ana is from Murcia and Luis is from Albacete."

- *Conjunction through shared components*: if there are modifications, repeated elements are looked for to only appear once:
  "Ana bought onions. Luis bought onions."
  "Ana and Luis bought onions."

- *Inclusion*: from the linguistic perspective, the most complex form of aggregation, involving subordinate sentences. The following example appear in (Cheng et al., 1997):
  "The house is near the bridge. The house is nice."
  "The house near the bridge is nice."

- *Lexical Aggregation*: whose objective is to express with a single term the meaning of a set of terms. It is also related to lexicalization, since the lexical component that will be substituted must be chosen:
  "It reacts with fluorine, chlorine, bromine and iodine."
  "It reacts with halogens."

In a NLG system the appropriate mechanisms will be selected to carry out these tasks. As in other stages of the system, it is necessary to consider aspects such as: the profile of the user ,which may require more or less complex texts; and the requirements of the system, having a limited space favors the conciseness of the text.

Lexicalization

Lexicalization is the task of the NLG that is in charge of selecting the specific words or concrete syntactic structures which to refer to the content selected in previous stages.

The variety of options that can be proposed for the same message can be both syntactic and semantic (Reiter & Dale, 2000). If the message to be conveyed is the scarcity of rain during a month, there are multiple possibilities:

- *Variations in the syntactic category*:

  Subject: "the rain was very scarce"

  Adjective phrase: "much drier that the average"

- *Semantic variations*:

  Absolute values: "very dry, very scarce rain"

  Comparative values: "some measures below average, much drier than the average"

When this happens and several options are available, aspects such as the knowledge and preferences of the user, the level of formality (e.g. "*father*" versus "*dad*"), consistency with both the lexicon already used and with the history of the discourse (e.g. contrast must be expressed if "*moreover*" already appears, "*in addition*" is used) or the relationship with the tasks of aggregation and GRE specific to this stage should be considered.

Generation of Referring Expressions

In certain linguistic contexts, the selection of one representation or another for a concept can generate ambiguity. Within the discourse, it must be possible to differentiate the entities and find the particular characteristics that contribute to satisfying the communicative goals. Determining the way in which the entities and concepts that form part of the document plan are referenced in order to avoid ambiguity is the function of GRE. Thus, following (Reiter & Dale, 2000), it will be assumed that the referring expressions must include the information that allows the unequivocal identification of a referent in the context of the discourse, avoiding redundancies or information overload.

The definition of the problem that occupies GRE is one of the most consensual in the NLG field.

The architecture being referenced in this work is sequential. This implies that the GRE must be carried out from the content selected in the first stage of the system. Therefore, when the selection takes place, it must establish what the text will need, considering that the form of the entity description depends on the place that it is in the context of the discourse. In this sense, it is usual to distinguish between the first allusion to an entity in the discourse (initial reference) and any other reference in the rest of the discourse, which will have to consider what has been said so far.

There is a possibility that the GRE asks the content selector what it needs to construct an adequate description. This would imply a bidirectional communication which is not possible in the sequential architecture exposed here. However, this is possible from another perspective, as presented in works such as (Hervás & Gervás, 2008), in which the generation of descriptions is guided by the GRE as opposed to the one guided by the content selection.

2.2.3 Surface Realization

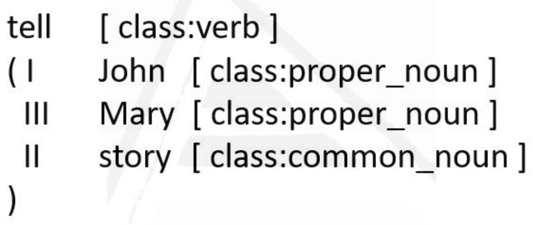
The tasks associated to the surface realization stage will have as a final objective the generation of real sentences that will form part of the output of the system, as well as structure and format of sentences, depending on the requirements of the application containing this stage. The syntax, morphology and orthography are aspects that are worked on in the tasks of this stage, which will finally produce a grammatically correct text that is susceptible of receiving the postprocessing which will give it the precise format.

The input for this stage is the specification of the text or the discourse plan, produced by the microplanning stage. This discourse plan is a set of specifications relative to sentences and their structure in the final discourse. The realization stage can be thought of as the set of tasks that translate those specifications into the output that the user receives.

In order to differentiate between the task that converts specifications into sentences and the one that gives them a format, two tasks are distinguished within this final stage: linguistic realization and the realization of the structure.

Linguistic Realization

Linguistic realization will determine how abstract representation of sentences becomes real text. The input for a tool called RealPro[3] (Lavoie & Rambow, 1997) is shown in Figure 2.4. The result in this case would be the sentence: "John tells Mary a story".



```
tell    [ class:verb ]
( I     John  [ class:proper_noun ]
  III   Mary  [ class:proper_noun ]
  II    story [ class:common_noun ]
)
```

Figure 2.4: Syntactic representation for RealPRO

Realization of the Structure

The last step of the NLG process is completely conditioned by the application. At this point the results of the previous steps will be formatted in a particular medium. The text may be shown in a webpage requiring HTML tags or, it may be a voice in a dialogue with a user. These are only two examples but there are multiple possibilities. Therefore, actions such as including tags in the document (e.g. HTML, LaTeX, RTF or SABLE[4]) or the creation of a tree including specific attributes to the final receiver (e.g. punctuation, comic strips, etc...).

2.3 Approaches to Address the NLG

In this section, an analysis of the most relevant approaches employed to address the NLG task is performed. The NLG task has traditionally been tackled from two perspectives: i) knowledge-based and ii) statistical.

[3] RealPro is a tool used in NLG focused on the surface realization stage. It will be further discussed in Section 3.1.1.

[4] SABLE is a XML markup language used to annotate text whose objective is voice synthesis.

First, we speak of knowledge-based systems when the techniques implemented are nourished from sources with a marked linguistic character, such as dictionaries, thesauri, lexical knowledge bases, rules or templates. From these resources morphological, lexical, syntactic and semantic information is extracted. Second, when a system is developed under a statistical approach, the information needed to transform the input into a text in natural language comes mainly from a corpus and the probabilities extracted from the texts which compose it, whether labelled or not. The latter systems are less restricted to a domain or to a language than those based on knowledge. This is because if the corpus employed is appropriate both in size and type of content, it does not have as many restrictions as those resulting from generating rules, which are limited to the characteristics of the context or to the peculiarities of a specific language. The restrictions and peculiarities of a specific language are common to the knowledge-based systems.

Apart from the aforementioned approaches, there are hybrid ones which combine statistical and knowledge-based techniques. This type of approach takes advantage of the strengths of both techniques and may overcome some of their weaknesses. In addition to this, recently there has been an increase in the use of deep learning techniques in the area of NLP. This type of system will be later discussed in Section 2.3.4.

2.3.1 Knowledge-based Approaches

Knowledge-based approaches have in common the ability to explicitly represent knowledge. For this purpose, these systems make use of tools such as ontologies, rule sets or thesauri.

These systems are considered to consist of two subsystems: i) a knowledge base and (ii) an inference engine. A knowledge base is a type of knowledge management database that provides the necessary means for the collection, organization and retrieval of knowledge. An inference engine is the part of the system that reasons using the content of the knowledge base in a given sequence. This engine examines the rules of the knowledge base one by one, and when the condition of one of these rules is met, the action specified for it is performed.

This systematization of the knowledge is based on linguistic theories that underpin the design and application of appropriate techniques. The most relevant theories are presented below, as they are the most used in the development of knowledge-based or hybrid systems.

- **Rhetorical Structure Theory** (RST) (Mann & Thompson, 1988) is one of the main theories employed in NLG and is related to the cohesion of the discourse as well as to the structure of messages and paragraphs. The idea behind RST is the possibility of recursively decomposing any text into a set of elements among which a series of rhetorical or discursive relationships, called schemes, are established. Examples of such relationships can be seen in Figure 2.5. The analysis of rhetorical relationships also considers the intentions of the person who starts the communication as well as the intended effects on the recipient. Some elements of the set are more relevant and are constituted as nuclei, while the elements that depends on them are referred as satellites. Two sentences can be also related and under the same scheme, their nuclei would be. In Figure 2.5, obtained from (Mann, 1999), a structure of a sentence based in this theory can be seen.

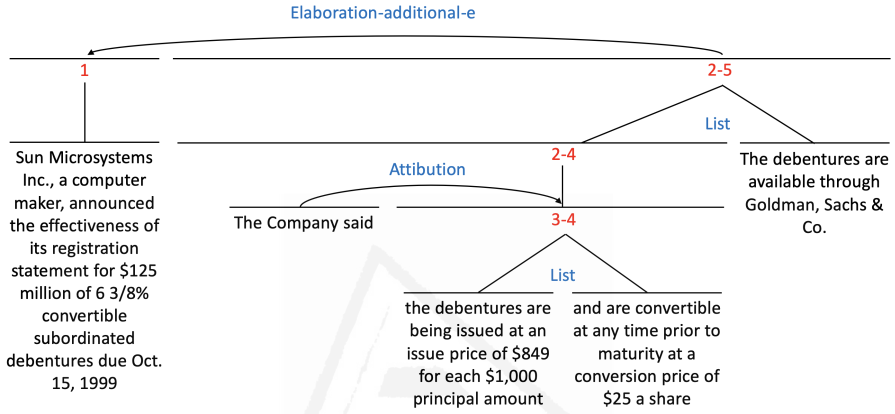

Elaboration-additional-e
1
2-5
List
Sun Microsystems Inc., a computer maker, announced the effectiveness of its registration statement for $125 million of 6 3/8% convertible subordinated debentures due Oct. 15, 1999
Attibution
2-4
The debentures are available through Goldman, Sachs & Co.
The Company said
3-4
List
the debentures are being issued at an issue price of $849 for each $1,000 principal amount
and are convertible at any time prior to maturity at a conversion price of $25 a share

Figure 2.5: RST Tree obtained from (Mann, 1999).

- **Systemic Functional Grammar (**SFG**)** (Halliday, 1985): For systemic functional linguistics, language is a resource which allows meaning to be constructed, and it is stratified into three levels: semantic, lexicogrammatic and phonological/graphological. SFG describes how the communicative functions can be expressed and affects the social dimension of language. This theory considers three dimensions: propositional, interpersonal (relations between the speaker and the recipient and how they influence the use of language) and textual (how information is structured and wrapped in a text). These last two metafunctions of language, in general, are not handled in other linguistic theories (Bernardos, 2003).

- **Tree-Adjoining Grammar (**TAG**)** (Joshi & Schabes, 1997): A TAG is a lexicalized grammar composed of a finite set of basic trees which incorporates semantic content. Starting from such trees, by means of substitution or adjoining unions, it is possible to construct a new tagged tree that represents the derivation corresponding to a sentence. One of the advantages of using a TAG is that it resolves in the same action the planning of the messages and their realization as a sentence (Koller & Petrick, 2011), although this entails a certain loss of flexibility. An example of the use of TAG can be seen in Figure 2.6.

- **Meaning-Text Theory (**MTT**)** (Mel'cuk et al., 1988): This theory uses a model of representation that differentiates the semantic, syntactic, mor-

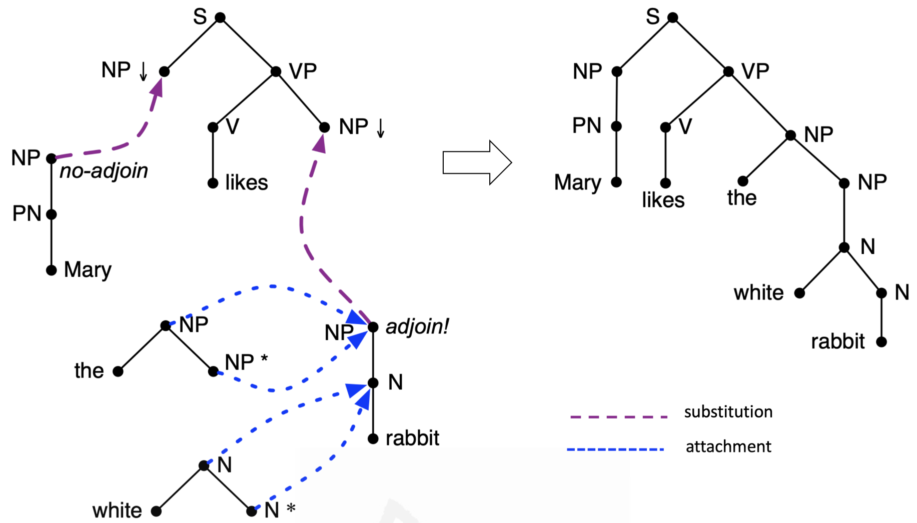


Figure 2.6: Example of derivation in a TAG extracted from (Koller & Stone, 2007). Applying substitution and attachment generates the derivation tree for the sentence “Mary likes the white rabbit”.

phological and phonetic levels. With the exception of the semantic level, the rest are divided into deep and superficial representations. According to this model, the NLG process would consist of the progressive transformation of the representations through the mentioned levels. Equivalence rules are used to perform the conversion from one level to another.

- **Centering Theory** (Grosz & Sidner, 1986; Grosz et al., 1995): The coherence of discourse and the way in which the entities composing it are related are addressed in the so-called models of discursive cohesion. Among them, the centering theory has been widely used in NLP to address the problem of anaphora, a linguistic phenomenon that occurs when elements of a sentence refer to entities that have already appeared in the discourse. According to this theory, an element of the discourse at a local level is constituted as the focus of attention or center of that context and is the most relevant entity referred by the rest of the utterances. Regarding the generation process, it affects, for example, the selection and use of pronouns and descriptions.

This type of technique has been used throughout the whole generation process. The system presented in (McDonald, 2010) performs the macroplanning using rhetorical operators, which after conducting an analysis of the communicative goals, expand the main communicative goal to reach a hierarchical tree structure in which the terminal nodes are the propositions and the operators are the rules of derivation thereafter. Regarding the microplanning, the aggregation task has been addressed employing a set of rules and information units that provide a unique output (Dalianis, 1996) or based on the exploration of dependency trees and on the RST (Theune et al., 2006). The surface realization stage has been interpreted from the MTT as a final step in a sequence of transformations carried out on linguistic representations, which can be tackled through grammar or rules that allow the translation of graphs (Wanner et al., 2010). Another example of a system employing these techniques is the one presented in (Gong et al., 2017). This system generates a news report with the use of knowledge rules and templates and the use of the tool described in (Zock & Lapalme, 2010). Moreover, the approach in (Perez-Beltrachini et al., 2012) is presented an approach for producing grammar exercises tailored to specific linguistic features, in the context of language learning. In this case, the surface realization stage is performed using the GraDe grammar traversal algorithm (Gardent & Kruszewski, 2012), where a sentences is generated based on a given grammar and a set of user-defined constraints.

2.3.2 Statistical Approaches

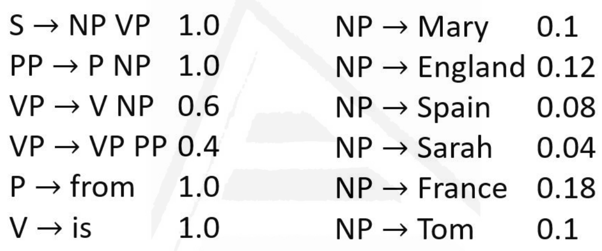
As mentioned before, statistical approaches are based on the probabilities extracted from a volume of text base, whether a corpus —annotated or not—, text from the Web, etc. Therefore, one of the big advantages of these approaches is their independence from language. Language Models (LM) are one of the primary tools for this type of approach.

A statistical LM is a mechanism that defines the structure of the language, that is, it adequately restricts the sequence of linguistic units based on a probability distribution that expresses the frequency of appearance for a sequence of n words $P(w_1, w_2, ..., w_n)$ in a set of texts. Thus, a good LM can determine, from the probability associated with a sentence, whether it is constructed correctly and if so, the sentence would be accepted by the LM. The sentence is rejected when the associated probability is low, indicating that such a sequence does not belong to the language on which the probability distribution was trained. It is important to highlight that a good LM can predict how an input (or part of that input) will be transformed within the NLG system or in one of its stages. One of the factors that determines the quality of a LM is the size of the corpus or data source from which the LM is trained, given that the number of use contexts of a word or the range of domains to which the LM can be applied will be proportional to the dimension of the training body. Three of the most commonly used LM in NLG are described below.

- **N-gram Model**: A n-gram is a subsequence of n elements of a given sequence. The n-gram model is a type of probabilistic model that enable a statistical prediction of the next element that will appear in a sequence of elements that have happened so far. These models can be defined by a

Markov chain[5] of order n-1. The implementation of these models is simple and is very useful in the construction of recognition and machine learning algorithms. However, n-gram models are very general, so it is necessary to adapt them to each application. A further limitation is that they are only capable of capturing relationships at short distances.

- **Models based on Stochastic Grammars**: Stochastic grammars are those in which each grammar rule has an associated probability, so that the result of applying the rules provides a probability that has been derived from them. Models based on stochastic grammars represent language constraints in a natural way. In addition, they permit the modelling of dependencies as long as necessary, although the definition of these models and their parameters is very difficult for complex tasks. Figure 2.7 shows an example of a simple stochastic grammar.



S → NP VP	1.0	NP → Mary	0.1
PP → P NP	1.0	NP → England	0.12
VP → V NP	0.6	NP → Spain	0.08
VP → VP PP	0.4	NP → Sarah	0.04
P → from	1.0	NP → France	0.18
V → is	1.0	NP → Tom	0.1

Figure 2.7: An example of a simple stochastic grammar.

- **Factored Language Models** (FLM): FLM, presented by (Bilmes & Kirchhoff, 2003), are an extension of the traditional LM. In this model, a word is viewed as a vector of k characteristics or factors, so that $w_t \equiv \{f_t^1, f_t^2, \ldots, f_t^K\}$. The factors within this model can take many forms, including lemma, stem, Part-of-Speech (POS) tag, or any other syntactic, lexical or semantic feature. The main objective of a FLM, based on the selected factors by the user, is to create a model $P(f|f_1,\ldots,f_N)$ where the prediction of a feature f depends on N parents $\{f_1,\ldots,f_N\}$. For example, if w represents a word token and t represents a Part-Of-Speech (POS), the expression $P(w_i|w_{i-2},w_{i-1},t_{i-1})$ provides a model for predicting the current word based on the traditional n-gram as well as the POS of the previous word. Consequently, in the development of these models there are two main issues to consider: 1) choosing an appropriate set of factors, and 2) finding the best probabilistic model over these factors. Figure 2.8 shows an example of this kind of models, extracted from (Kirchhoff et al., 2007), in the form of a dependency directed graph.

[5] A Markov chain is a special type of discrete stochastic process in which the probability of an event occurring solely depends on the immediately preceding event.

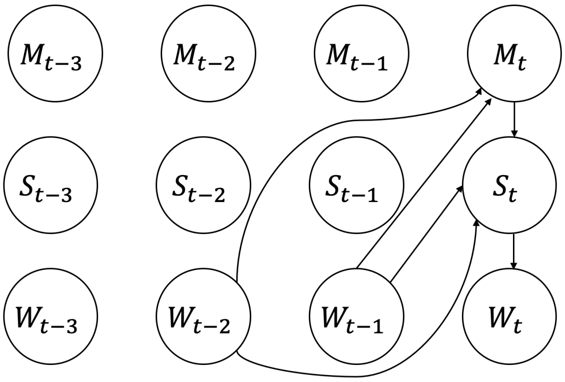


Figure 2.8: Example of a FLM, which is seen as a directed graph, with words W, morphological factors M and stems S as factors. The figure shows that in this model, a word W_t is determined by the stem S_t and the morphological factor M_t.

This type of technique is not usually used in all of the aforementioned stages that constitute the architecture of a NLG system, but they do in the last two stages of the process. For instance, the system presented in (Ballesteros et al., 2014) addresses the lexicalization task by developing a statistical generator that is capable of selecting the terms corresponding to a set of semantic representations by means of classifiers (i.e. Support Vector Machines), based on the AnCora-UPF treebank (Mille et al., 2013). The task of GRE has been also tackled from a statistical perspective, where the system mCRISP (Garoufi & Koller, 2011) generates referring expressions using classifiers trained over a corpus of descriptions. Regarding the surface realization stage, one of the first works that presented corpus-based statistical techniques was the one developed by *Langkilde and Knight* in 1998 (Langkilde & Knight, 1998) whose system used these statistical techniques in the previous stages as well. A n-gram model was used which, as far as the stage of realization was concerned, determined word transformations (whether to use plural or not, gender, etc.). Those with the highest probability were selected to appear in the final output of the system. On the other hand, in (Barros & Lloret, 2017), the surface realization stage is performed using FLM to generate sentences for different domains.

2.3.3 Hybrid Approaches

Hybrid approaches are those that combine knowledge-based techniques and statistics to perform several tasks that fall under the NLG. Since the end of the 20th century, works employing this type of approach can be found. FERGUS (Flexible Rationalist-Empiricist Generation Using Syntax) (Bangalore & Rambow, 2000) was one of the first hybrid systems created for NLG, which performed the microplanning and surface realization stages. This system combines N-grams models with a tree-based statistical model and uses a lexicalized tree-based syntactic grammar, which is based on XTAG grammar, to generate text. The application FLIGHTS (White et al., 2010) is another example of such a hybrid system. It presents flight information in a personalized way for each user (e.g. considering whether a user is a student or a frequent flyer). In its development different knowledge bases are considered (user models, domain models and dialog record) to decide the content that should appear in the output. This content is then structured from templates and the final text is generated employing the OpenCCG framework[6]. This tool internally uses n-grams models and FLM.

More recently, Kondadadi et al. (2013) presented a hybrid of statistical and template-based (which are used in many knowledge-based approaches to generate text) systems that consolidates the three stages of the NLG pipeline into one statistical learning process. This system first automatically derive a template bank from a corpus for a given domain. Then, they use a statistical ranking model to select the best template that fits the system input data. In (Mille et al., 2016), a preliminary proposal of a multilingual system for abstractive summarization using semantic representations is described. In this proposal, the system, whose underlying theoretical framework would be the MTT, would combine statistical and rule-based techniques to produce a summary in response to a user query. Gardent and Perez-Beltrachini (2017) proposed a hybrid symbolic/statistical approach to model the constraints for regulating the fine-grained interactions between the tasks of a NLG system. This approach uses a small handwritten generic grammar, a statistical hypertagger and a surface realization algorithm to fulfill this purpose. Regarding Spanish, in (García-Méndez et al., 2018) a hybrid system to generate sentences from pictograms is proposed. This system combines linguistic knowledge given by a lexicon and a language model to infer prepositions. Then this knowledge, in conjunction with an adaptation to Spanish for SimpleNLG (Gatt & Reiter, 2009) (this tool will be detailed in Section 3.1.1 of Chapter 3) is used to generate coherent sentences.

2.3.4 Deep Learning Approaches

In recent years, deep learning approaches have gained popularity throughout the NLP area. Likewise, in the NLG field some works have arisen in the last two years. To the best of our knowledge, this type of technique is not widespread enough within NLG. Therefore, the number of existing works using this technology is lower than those that can be found using classical approaches (e.g. knowledge-based and statistical approaches).

In this regard, an example of a system using these techniques can be found in (Lebret et al., 2016), where a neural model for D2T is proposed. This model, which is built on conditional neural LM, generates biographical sentences from Wikipedia biographies. In (Brad & Rebedea, 2017) a neural paraphrase approach is presented. This approach employs sequence-to-sequence models with at-

[6] http://openccg.sourceforge.net/

tention, in conjunction with transfer learning, and uses textual entailment and phrasal paraphrase pairs for the generation of paraphrases. Recently, Castro Ferreira et al. (2018) presented an approach for GRE which relies on deep neural networks. This approach makes decisions about the form and the content of the generated text without performing feature extraction explicitly.

Concerning the deep learning systems, for the NLG task, its outputs may contain incorrect content or add content which is not explicitly in the input[7]. This may not be adequate in certain NLG applications, such as the generation of medical or financial reports, where the desired information generated has to be trustworthy and accurate. There are also other cases where the generated language is ungrammatical, and the content is meaningless (Subramanian et al., 2017).

2.4 Conclusion

The second chapter presented the state of the art in NLG that describes the most important issues when developing a system in this field. A brief review of the classification, the architecture and the techniques employed to approach the NLG was conducted to provide the necessary background information on this research area.

The analysis performed in this chapter provided a round-up of the state of the art of NLG. As for the types of systems, it is worth mentioning that there is a wide range of areas where the NLG can be applied. Ranging from the text summarization to the text simplification, these areas can benefit from what the NLG can offer to improve their results. Despite no clear consensus on how the architecture of generation systems should be designed, there is a strong tendency to use the one proposed by Reiter and Dale (2000).

Two types of techniques predominate for approaching the different tasks of a NLG system. On the one hand, the knowledge-based approaches rely on a linguistic background. However, these techniques are too rigid and usually depend on language and domain. On the other hand, the statistical approaches can surpass these types of restrictions, but cannot bring as much information to the development of a system compared to the knowledge-based ones. Therefore, the hybrid approaches that combine these two techniques may provide more flexibility when developing a system in terms of language or domain.

Although NLG's origins date back to the mid-twentieth century, there is a great interest among the research community in this field due to its adaptability to different applications and what NLG can offer to other NLP fields. Therefore, acquisition of a sound background in this field is essential.

Considering what has been discussed in this chapter, there is still a lot of room for improvement in this research field. In this sense, the development of versatile NLG approaches is still a challenge. Existing NLG systems are usually

[7] https://ehudreiter.com/2018/11/12/hallucination-in-neural-nlg/

developed for concrete purposes, domains and languages, and their adaptation to other domains or languages is very costly. Research into more flexible and easily adaptable approaches, for other domains or languages, would certainly be a breakthrough in the NLG area.

CHAPTER 3

Resources and Evaluation in Natural Language Generation

In the previous chapter, a wide review was carried out of the techniques employed in NLG as well as the type of NLG systems and its architectures. It is also important to know the available resources in this research field. Hence, the key to developing new approaches includes the tools and datasets, as well as how to evaluate the output of NLG systems, or knowing where the newest work in this field is published. Therefore, the main objective of this chapter is to provide the necessary and relevant review of the current available resources for the NLG research field.

The structure of the chapter is as follows. Section 3.1 describes the available resources for NLG, ranging from the tools performing the whole generation process or only one of the stages separately, to the corpora specifically created for the generation task. Section 3.2 explains how the NLG systems have been evaluated from different perspectives. Section 3.3 provides a summary of the most important conferences focused on NLG and a brief review of the general NLP conferences. Finally, Section 3.4 concludes the chapter by providing some insights on future directions of NLG.

3.1 Tools and Corpora

In Section 2.2 of Chapter 2 the different stages that constitute a generic NLG system were described. A series of tools used in these stages will be subsequently analyzed.

There are many free tools and applications on the Web, although they are mainly focused on the realization stage. Numerous systems and resources are available on the web of *Bateman and Zock*[1], and abundant documentation re-

[1] http://www.nlg-wiki.org/systems

lated to techniques, underlying theories, evaluation and new challenges can be found in the proceedings of both the International Natural Language Generation Conference (INLG) and the European Natural Language Generation Workshop (ENLG) (see Section 3.3).

3.1.1 NLG tools

The following are a set of tools that have been selected for their relevance or for being associated to each of the stages. These tools will be described based on the stages they tackle. Therefore, we start first with the tools that only deal with one stage of the NLG pipeline, and we finish with the ones addressing all stages. In Table 3.1 a summary of these tools is shown.

Macroplanning

SPUR (Walker et al., 2007) is responsible for the macroplanning stage. This tool is used in a system that recommends or compares options (e.g., restaurants), and this determines the structure it produces. Therefore, the input of this system are the attributes of the options to be compared and the output is a document plan with the most important attributes of the options. These attributes are the most relevant for the user and are based on their preferences and requests. Depending on the training, SPUR can produce different content plans.

This system was specifically created as part of the system MATCH (Johnston et al., n.d.), a tool for comparing and recommending restaurants in New York. So, the adaptation of this tool to be use jointly in another system would be complex, since it was developed ad hoc.

Microplanning

SPaRKy (Walker et al., 2007) is in charge of the microplanning stage based on a template system. The input to this system is the document plan generated by SPUR, and the output of this tool is a discourse plan with the assertions that were represented at input. The process is divided into two phases or modules:

- Sentence Plan Generator. In the first phase, a set of planning trees is generated containing the relationships (internal nodes) and the assertions (leaf nodes) that appear in the final text. In the second phase, those assertions are assigned to sentences and then organized.

- Sentence Plan Ranker. Evaluates the different plans generated by the Sentence Plan Generator relying on a model based on user ratings in the training phase.

SPaRKy is trained by using user feedback, and because of that it could be used as an individualized spoken language generator. However, it uses some domain-dependent knowledge to deal with the planning trees, so its adaptation to other domains would be also complex.

Tool	Stages	Input	Output
SPUR (Walker et al., 2007)	Macroplanning	Attributes that the object to be compared has.	Document plan with the most important attributes.
SPaRKy (Walker et al., 2007)	Microplanning	Document plan (generated by SPUR).	Discourse plan with assertions to be represented at the output.
mSCRISP* (Garoufi & Koller, 2011)	Microplanning	A corpus of human-generated instruction giving sessions.	A well formed referring expression.
RealPRO (Lavoie & Rambow, 1997)	Realization	Discourse plan (nodes such as lexemes and edges such as syntactic relations) (D2T).	Syntactic and sematically correct sentences.
SimpleNLG (Gatt & Reiter, 2009)	Realization	Syntactic sentence structure (D2T).	Syntactic and sematically correct sentence.
PESCaDO (Wanner et al., 2012)	Macroplanning Realization (including Microplaning processes)	Multilevel ontology and user request.	Report as well-formed discourse.
PASS* (Van der Lee et al., 2017)	Macroplanning Realization	Match information from Goal.com.	Report as well-formed discourse.
NaturalOWL (Androutsopoulos et al., 2013)	Macroplanning Microplanning Realization	OWL Ontology (D2T).	Syntactic and sematically correct sentence.
BabyTALK (Gatt et al., 2009)	Macroplanning Microplanning Realization	Signal from medical devices.	Syntactic and semantically correct text.
FLIGHTS* (White et al., 2010)	Macroplanning Microplanning Realization	An abstract communicative goal from the system's dialogue manager.	Syntactic and semantically correct text speech.

Table 3.1: NLG tools. The marked systems have been mentioned along Chapter 2 to illustrate several examples of the different types of approaches and systems to address the NLG.

Surface Realization

- RealPRO (Lavoie & Rambow, 1997) is a tool which executes the realization stage and whose final output is a set of well-formed sentences in *ASCII*, *HTML* or *RTF* format. The tool input is a discourse plan structured in a diagram in the form of a dependency tree. The system processing is carried out on a linguistic knowledge base, initially only in English, but expandable to other languages. The input has two components:

- Syntactic relationships, which are represented with labels in the arcs that relate the nodes.
- Lexemes, which are represented with a label on each node. In addition, only the lexemes that provide meaning are stored. The tool is not capable of performing a syntactic analysis, so all the lexemes that provide meaning have to be specified.

Once the tree is constructed, the tool is responsible for adding the functional words, thus generating a second tree. With this second tree, based on the labels of the arcs, rules of linear precedence are created. These rules are later used for the inflection of the elements of the sentence. Finally, punctuation marks are added, and the necessary instructions are generated to adapt the output to the selected format. In figure 3.1 is shown an example of an input with its output.

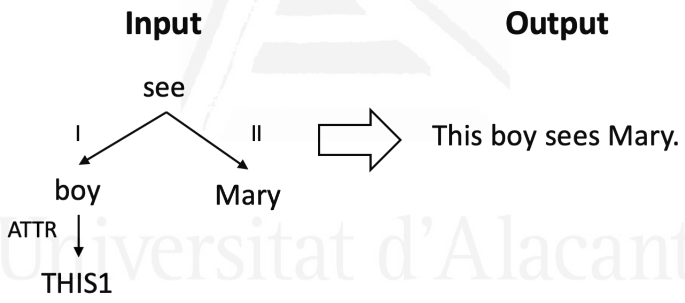


Figure 3.1: Example of an input and and output of RealPRO.

- SimpleNLG (Gatt & Reiter, 2009) is a tool focused on the surface realization stage which has been initially developed for English, but some versions for other languages have been recently released. In this regard, there are versions of this tool for the following languages: Spanish (Ramos Soto et al., 2017), French (Vaudry & Lapalme, 2013), German (Bollmann, 2011) and Italian (Mazzei et al., 2016). The tool can be found in the form of a library, written in the Java language, whose objective is to assist the writing of grammatically correct sentences. The input of this tool is a syntactic sentence structure and its output is a well-formed sentence. The tool has been built on three basic principles:
  - Flexibility. SimpleNLG is a combination of canned system (based on schemes) and advance system. By combining the two, a greater syntactic coverage is achieved.

- Robustness. When an input is incomplete or erroneous, the tool will generate an output, even though it is unlikely to be the expected one.
- Independence. Morphological and syntactic operations are clearly differentiated and separated.

The library provides an interface with which to interact from the Java code. Starting from a base element, which is equivalent to the main verb of the sentence, other elements that will take part in the main action are concatenated. Once the elements are grouped, the verb tense of the sentence and the way in which it is constructed (e.g. interrogative, infinitive,...) will be indicated. Finally, the tool generates a sentence based on the parameters configured. An example of an input and an output can be found in Table 3.2. The main advantage of this system is its simplicity and its ease of use, however, in order to generate a sentence, all the elements of the sentence that provide relevant information are required.

Verb	Parameters	Mood, "Output"
"*Leave*"	tense = past	Interrogative (where, object): "*Where did the boys leave?*"
	object = "*the house*"	Interrogative (yes, no): "*Did the boys leave the home?*"
	subject = "*the boys*"	

Table 3.2: SimpleNLG input and output example.

Several Stages of the Pipeline

- PESCaDO (Wanner et al., 2012) is a project developed to provide environmental information tailored to the profile of the user, preferences and location. This system generates a report from a basic environmental knowledge base that combines data from web services as well as information related to other user profiles. The input to the system is a multilevel ontology and a user request; and its output is a well-formed report. This system only performs the macroplanning stage and the surface realization stage.

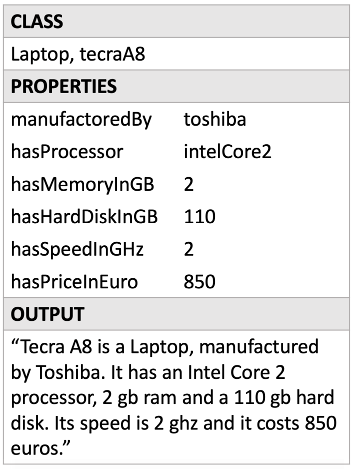
  - Macroplanning: the content selection is carried out based on the request of the user that gives rise to the population of a multi-level ontology. The nodes that constitute it will be grouped by themes, so that messages can be extracted as elementary units of the discourse. These units can be associated to a set of schemes that will determine its structure.
  - Surface realization: this stage transforms the abstract structure of the macroplanning into a suitable output which takes as its theoretical basis the MTT (see Section 2.3.1 of Chapter 2). The process consists

of mapping the different adjacent linguistic structures using a set of transition rules for each transformation level, using a tool called MATE (Wanner et al., 2010).

- NaturalOWL (Androutsopoulos et al., 2013) is a D2T tool that generates a text from an OWL ontology with the information contained in it. Therefore, the input to this system is an OWL ontology. OWL is a standard for specifying ontologies in the Semantic Web. To generate the text, this tool performs the three stages discussed in Section 2.2. The final output of the system is a syntactically and semantically correct sentence.

  - Macroplanning: The tool collects all the ontology statements that are considered relevant and converts them to a simpler format (triplets). Subsequently, the tool selects which triplets will be in the text, and an attempt is made to convert each one of them into a simple sentence. To do this, the triplets are ordered rather than the corresponding sentence. This is because NaturalOWL does not take into account the overall coherence since most sentences only provide additional information to the core or, at most, to the second level core. For the latter, the tree representation, which is usually used at this stage, is not employed.
  - Microplanning: NaturalOWL permits the user to set the maximum number of sentences to add. NLG systems generally add as many sentences as possible to improve the readability, but this tool permits the configuration of the maximum number of sentences to be concatenated.
  - Surface realization: NaturalOWL takes the output from the microplanning and represents it by adding the necessary punctuation marks and capital letters. The input of this last stage, as in most schema-based systems, contains the format and the final order in which each of the words will appear in the final text, so there is no need to add new information. Therefore, it is more a process of transforming the data obtained to the output format than a surface realization stage itself. An example of an input and output in NaturalOWL is illustrated in Figure 3.2. This system is domain-dependent so its adaptation to other domains may be complex since it may need the creation of domain-dependent generation resources for a specific target domain.

- BabyTALK (Gatt et al., 2009) is a system that generates reports on the status of neonatal patients depending on the type of user they are aimed at (doctors, nurses, parents, etc.). This system is based on the three stages pipeline architecture previously explained with the addition of two data preprocessing stages. The input to the system are signals from medical



CLASS	
Laptop, tecraA8	
PROPERTIES	
manufactoredBy	toshiba
hasProcessor	intelCore2
hasMemoryInGB	2
hasHardDiskInGB	110
hasSpeedInGHz	2
hasPriceInEuro	850
OUTPUT	
"Tecra A8 is a Laptop, manufactured by Toshiba. It has an Intel Core 2 processor, 2 gb ram and a 110 gb hard disk. Its speed is 2 ghz and it costs 850 euros."	

Figure 3.2: Example of the performance of NaturalOWL

devices (such as a heart rate monitor) and the output is a report about the status of the neonatal patients.

- Preprocessing stage: an analysis of the input signal is performed. This analysis will result in the identification of medically significant events and short and long-term patterns or trends.
- Data interpretation stage: starting from the identification obtained in the previous stage, a set of common events are grouped into a higher level and then interpreted. As a result of this stage, the interpreted events are obtained.
- Macroplanning: The events from the previous stage are ordered forming a tree of events which is the output of this stage.
- Microplanning and Surface realization: These two stages are carried out together. The tree of events of the previous stage is first converted into an event structure to which diverse concepts will be added, such as the linking of events to each other. Finally, starting from this structure, the final text is generated.

Example 7 shows a snippet of a text generated by BabyTALK.

(7) **Background**
The baby was born at 24 weeks weighing 460g. He is 2 days old and in intensive care.
Respiration
Current Status
The baby is currently on CMV. Ventilator BiPAP rate (vent RR) is 55 breaths per minute. Pressures are 20/4. Inspired oxygen (FIO2) is 27%. Ventilator tidal volume is 1.5. The most recent blood gas was taken 11 minutes ago. Parameters are normal. Ph is 7.3. Concentration of carbon dioxide (CO2) is 5.72kPa. A suction was done. There were blood stained secretions and purulent secretions.

> *Events During the Shift*
>
> An ABG was taken at 23:09. There was evidence of respiratory acidosis. [...] The baby was moved from BiPAP to CMV. He had been intubated.
>
> [...] Another ABG was taken in the early morning. There was evidence of respiratory acidosis. Ph was 7.18. CO2 increased to 8.74kPa.
>
> Blood gas parameters had improved by 06:28. [...] The last blood gas was taken 11 minutes ago. Ph increased to 7.3. CO2 dropped to 5.72kPa.
>
> *Potential Problems*
>
> Purulent secretions during shift suggest risk of infection.

3.1.2 Corpora

The use of corpus is common for certain NLG system development strategies, either intrinsically in the generation process (Section 2.3) or extrinsically in the evaluation process (Section 3.2). The corpus employed can be tagged with information of a different nature and will be selected depending on the task to be solved. For instance, the type of information included in a corpus necessary in the content selection stage is different from that required in the GRE stage. It is possible to find systems that use general corpus and systems that employ corpus specifically designed for NLG. The latter are usually created ad hoc for a particular application or within a competition that involves solving a very limited task (see Section 3.2.5).

In Table 3.3 a set of corpus specifically created for NLG is shown. They have been classified according to the tasks described in Section 2.2. These corpora are all in English and the information they present may be disparate. On the one hand, the corpora oriented to content selection and aggregation contain sets of data, both numerical and textual, from which information can be selected for further processing. On the other hand, corpora oriented to generate referring expressions contain information about real objects or referring expressions themselves. Finally, a corpus is included that is used in the surface realization stage. The main characteristic of this type of corpora is that they contain structured data.

[2] http://inf.abdn.ac.uk/research/sumtime

[3] http://www.classic-project.org/

[4] http://jetteviethen.net/research/spatial.html.

[5] http://inf.abdn.ac.uk/research/tuna/corpus

[6] http://www.pitt.edu/~coconut/coconut-corpus.html

[7] http://mcs.open.ac.uk/nlg/old_projects/pills/corpus/PIL/

[8] https://gitlab.citius.usc.es/alejandro.ramos/geodescriptors

[9] https://github.com/muskata/SpatialVOC2K

[10] http://webnlg.loria.fr/pages/docs.html

[11] http://www.macs.hw.ac.uk/InteractionLab/E2E/

[12] https://github.com/harvardnlp/boxscore-data

[13] https://github.com/DavidGrangier/wikipedia-biography-dataset

Corpus	Information	Type of info.	Tasks
SumTime [2]	Predictions of meteorological parameters	Data (D2T)	Content Selection and Aggregation
WOZ[3]	Selection of attributes of an Edinburgh restaurant	Text (T2T)	Content Selection and Aggregation
GRE3D7[4]	Referring expressions about 3D objects	Data (D2T)	Generation of Referring Expressions
TUNA[5]	References to objects in visual domains	Data (D2T)	Generation of Referring Expressions
COCONUT[6]	Automated dialogues	Data (D2T)	Generation of Referring Expressions
PIL[7]	Patient information	Data (D2T)	Realization
Geodescriptors[8]	Geographical descriptors and a set of associated graphical representation	Data (D2T)	Generation of Referring Expressions
SpatialVOC2K[9]	Images with annotations and features for spatial relations between objects	Data (D2T)	Generation of Referring Expressions
WebNLG[10]	DBpedia RDF triplets	Data (D2T)	The whole NLG process
E2E[11]	Information about restaurants	Data (D2T)	Sentence Planning and Surface Realization
ROTOWIRE[12]	NBA summaries and scores	Data (D2T)	The whole NLG process
WikiBio[13]	Biographies from Wikipedia	Data (D2T)	The whole NLG process

Table 3.3: NLG corpus.

3.2 Evaluation in NLG

There is a general consensus among scholars on the difficulty involved in the NLG evaluation task due to its peculiarities (Viethen & Dale, 2007). Compared to other systems developed in NLP, in NLG the evaluation of the system will be carried out considering that, to begin with, what should be the input to the system and the stages is not adequately specified. Furthermore, the correct output is not unique and there is no defined criterion to evaluate output quality.

3.2.1 Types of Evaluation

When a NLG system is evaluated, different strategies can be followed (Resnik & Lin, 2010). First, it is possible to evaluate the impact of the system on users or other tasks. This is the case of an extrinsic evaluation, which is focused on the external effects on the system. Second, the performance and effectiveness of the system itself can be evaluated, in which case an intrinsic evaluation would be carried out. A distinction is also made between the manual and automatic evaluation. The former is usually more expensive and more difficult to organize and may even take long time to complete.

In past NLG systems, intrinsic evaluation has been commonly addressed using human assessors (manual evaluation), who have being involved in reading and rating texts or comparing the ratings for texts generated by a NLG system (Gkatzia & Mahamood, 2015). In addition, the intrinsic evaluation of text quality has been approached through the use of automatic metrics (e.g., ROUGE, BLEU, etc.) (Reiter & Belz, 2009).

Conversely, extrinsic evaluation can include diverse aspects to assess such as: measuring the correctness of the decisions made in a task based evaluation; measuring the number of post-edits by experts; and measuring the utility of the system. Therefore, this type of evaluation can be either performed manually or automatically. For example, an extrinsic and manual evaluation was performed for the STOP system (see Section 2.1.2). Surveys were used to monitor the effectiveness of the tasks: how many users had quit smoking, how long it took, etc. The evaluation of STOP took 20 months and cost 75,000 pounds (Reiter et al., 2003).

3.2.2 Relevant Aspects in the Evaluation of NLG Systems

The task of evaluating a NLG system includes many aspects that need to be delimited and defined. Consideration can be given to aspects related to the whole system performance, as well as to the performance of each of the system stages (Reiter, 2010).

The evaluation of the system takes into account the adaptation of the output to the communicative goal, to the history of the discourse or to the request of the user. It also takes into account the syntactic coverage and style correction, as well as coherence, ambiguity and quality of the vocabulary (Bernardos, 2003). Sometimes the effort required to post-edit the output is measured or experiments are carried out with users who complete reading and comprehension tests, or have to score the output, for example.

Regarding the performance of each of the stages or tasks of the NLG system, each stage must be evaluated considering its own responsibilities (Bernardos, 2003):

- *Content selection*: quality of the information displayed
- *Structuring the document*: cohesion
- *Sentence aggregation*: cohesion and redundancy
- *Lexicalization*: quality and coverage of vocabulary

- *Generation of Referring Expressions*: quality of the information, ambiguity and redundancy
- *Linguistic realization*: syntactic coverage, fluency and clarity
- *Realization of the structure*: effort to post-editing the output, legibility and clarity

3.2.3 Metrics to Evaluate NLG Systems

The possibility of evaluating the output of a NLG system by comparing it with an ideal text created by an expert or with a reference corpus, whether generated by humans or by other NLG systems, has been discussed above. The evaluation in this case can be done in quantitative terms using metrics made on such comparisons. The metrics that are usually used come from other NLP areas and have been adopted due to their good results in their respective fields. This type of automatic corpus-based evaluation is attractive in NLG, as in other areas of NLP because of its speed, reproducibility and low computational cost (Reiter & Belz, 2009).

Some of the metrics employed in NLG come from the machine translation field, such as BLEU (Papineni et al., 2002), NIST (Doddington, 2002) or METEOR (Lavie & Agarwal, 2007). Others, such as ROUGE (Lin, 2004) come from the summarization field. BLEU is a precision metric used in machine translation that evaluates the proportion of n-grams that share the system output with multiple translations. NIST is an adaptation of BLEU that adds a weight to the most informative n-grams. METEOR is a metric which, apart from comparing the n-grams, compares meanings from sources such as WordNet [14] (Fellbaum, 1998). In the summarization field, although there are other metrics, the ROUGE tool is the most used and has a parallel performance to BLEU. This tool provides metrics (e.g. ROUGE-1, ROUGE-2 or ROUGE-SU4) to evaluate how informative a summary is by comparing the n-grams to one or more model summaries. The main difference between BLEU and ROUGE is that in the latter, the recall is oriented.

These strategies are discussed in the NLG community due to diverse factors. In NLG there is more than one good gold standard output, whereas other NLP fields only have one (e.g., opinion mining, question answering, etc.). Nor are there many specific corpora for the task and the metrics employed come from other NLP areas (Scott & Moore, 2007). Moreover, normally scholars point to the difficulty of interpreting the results provided by the metrics (such as, what a variation of the results implies, how to compare the results with human evaluation, etc.) (Paris et al., 2007). These are some of the reasons given for distrusting this type of evaluation. However, the subject remains open and under constant review. Diverse studies can be found, such as the one carried out in (Belz & Reiter, 2006),

[14] WordNet is a lexical database for English

where after using the mentioned metrics (i.e. NIST, BLEU and ROUGE) on the output of SumTime and comparing the results with human evaluations, it was concluded that it is appropriate to make use of the metrics in NLG, albeit under certain conditions if good results are expected. Hence, a broad and high-quality corpus is needed but metrics that allow the evaluation of certain linguistic aspects of the text, that go beyond the mere comparison of n-grams (e.g. the structure of the information), are also needed.

3.2.4 Manual Evaluation in NLG

The use of automatic methodologies (e.g., such as the metrics previously mentioned —ROUGE or BLEU—) to evaluate a NLG system may not be enough when assessing some aspects of a generated text, such as its meaningfulness or its correctness. Therefore, a manual evaluation would be more appropriate in these cases. The evaluation based on human ratings and judgements is currently one of the most used within the NLG field (Reiter & Belz, 2009). In this type of evaluation, human subjects are usually asked to evaluate texts on surveys, questionnaires or crowdsourcing platforms. These type of surveys are generally composed of several questions which depend on what aspects of the text are evaluated.

The questions asked in a human rating evaluation differ from the ones in a human judgement evaluation. In the former, human assessors are asked to evaluate the linguistic quality (i.e., readability or fluency) and the content quality (i.e., accuracy, adequacy, relevance or correctness) of the text using rating scales. The most used scale in these cases is the 5 point Likert-scale — strongly agree (5), agree (4), undecided (3), disagree (2), strongly disagree (1)—, where the assessors have to rate from 1 to 5 the aspects asked. For instance, in (Mitchell et al., 2012) the output of their system assess different aspects of the text (i.e., grammaticality, correctness and humanlikeness — if the text seems to have been written by a person—) using the crowdsourcing platform Amazon's Mechanical Turk[15]. In (Espinosa et al., 2010) two human judges were asked to evaluate the adequacy and the fluency of their system's output with 5 pt-Likert scale.

When performing a human judgement evaluation, assessors are asked to order, based on their preference, several texts (Belz & Kow, 2010). In this regard, in (Reiter et al., 2005b) the human evaluators were shown two distinct variants of forecast texts and were asked to decided which one was easier to read, more accurate and more appropriate.

The main concern of human-based evaluation is the subjectivity inherent in each human assessor which may affect inter-rater reliability. This may lead into a high variance in the judgements made by several assessors (Gatt & Krahmer, 2018). However,there are crowdsourcing platforms that can calculate "trust" for participants. One of the examples is ARGO system (Pittaras et al., 2019) for the evaluation of the single-document summarization.

[15] https://www.mturk.com/

3.2.5 Collaborative Evaluation and Competitions

Determining how systems attached to a particular discipline are evaluated is considered to be a crucial aspect for advancing the research and progressing the discipline. The debate on NLG dates back to the last decade of the 20th century, when it began to differ from the other areas of NLP (Mellish & Dale, 1998). However, more recently a greater effort has been made to define the appropriate methodology in evaluation and initiatives have been launched to determine common reference frameworks and provide suitable spaces to discuss the evaluation issue. In response to the growing interest in this field, a first special session was held in 2006 at INLG (INLG'06 Special Session on Sharing Data and Comparative Evaluation). In that special session, the groundwork was laid for new projects focused on evaluation in NLG, anticipating the following meetings[16]. The creation of an organization whose mission would be to promote competitions related to different tasks of NLG impacted the way in which the systems are evaluated. The group in question was called *Generation Challenges* and its research has resulted in what is known as Shared Task Evaluation Challenges (STEC) in the context of NLP, i.e. collaborative evaluation research based on the approach of a specific problem, referred to a NLG task, whose resolution has to be faced by several work teams, finally comparing the results obtained (Viethen & Dale, 2007). The so-called *challenges*.

Although the implementation of such competitions is quite recent, the variety of challenges proposed has given rise to very diverse contentions. A list of some of them is shown in Table 3.4.

However, the use of STEC in NLG has been discussed from the moment they appeared (Dale & White, 2007). Due to the complex nature of this discipline some issues must be considered including: the type of tasks to be evaluated by these methods; the type of metrics to be used; and, the necessary methodological bases (both in terms of the approach of the competition and the comparison of the results). Some alternatives are also considered. According to (Walker,

[16] Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation 2007, Workshop: Using Corpora for Natural Language Generation + Evaluation 2011

[17] http://taln.upf.edu/pages/msr2018-ws/SRST.html

[18] http://www.macs.hw.ac.uk/InteractionLab/E2E/

[19] http://alt.qcri.org/semeval2017/task9/

[20] http://talc1.loria.fr/webnlg/stories/challenge.html

[21] http://www.taln.upf.edu/cschallenge2013

[22] https://sites.google.com/site/hwinteractionlab/current-research/generation-challenge

[23] http://www.taln.upf.edu/cschallenge2013

[24] http://www.kbgen.org

[25] https://www.mq.edu.au/research/research-centres-groups-and-facilities/innovative-technologies/centres/centre-for-language-technology-clt/research/projects/hoo-helping-our-own

[26] http://www.questiongeneration.org

[27] http://www.nltg.brighton.ac.uk/research/sr-task

Year	Competition	Objetive	NLG stages
2018	*SR-18*[17] *First Multilingual Surface Realisation Shared Task*	Generation of sentences from syntatic and semantic representations.	Surface Realization [*Re*]
2018	*E2E NLG Challenge*[18]	Generation of utterances from a given dialogue act-based meaning representation.	Entire process [*Mi+Re*]
2017	*SemEval-2017 Task 9*[19]	Generation of English sentences from AMR triplets.	Surface Realization [*Re*]
2017	*WebNLG Challenge*[20]	Generation of sentences from RDF triplets.	Entire process [*Mi+Re*]
2013	*GIVE*[21]	Construction of a system that generates orientation and manipulation instructions for a user to move into a virtual building and to obtain a trophy.	Entire process [*Mi*]
2013	*GRUVE*[22] *Generation of orientation instructions in uncertain virtual environments.*	Construction of a module to help navigation in Google Streetview.	Entire process [*Mi*]
2013	First challenge in content selection from the open semantic web[23]	Construction of a system that recovers, from a set of RDF triplets referred to a celebrity, those that appear implicit in a target text.	Content selection [*Ma*]
2013	*KBGen*[24] *Generation from Knowledge Base*s	Generation of a coherent description of biological entities, processes and connections between them.	Entire process [*Ma+Mi+Re*]
2012	*HOO*[25] *Helping Our Own*	The purpose of this task was to improve the quality of English texts, especially those written by non-native speakers.	Entire process [*Ma+Mi+Re*]
2011	*GQ*[26] *Question Generation Shared Task*	Generation of questions from sentences and paragraphs using Wikipedia, OpenLearn and Yahoo! Answers as sources.	Entire process [*Ma+Mi+Re*]
2011	*SR-11*[27] *Realization of the Structure*	Generation of sentences from syntatic and semantic representations.	Surface Realization [*Re*]

Table 3.4: Competitions in the field of NLG. These tasks are developed either focused on solving a specific stage or on generating a complete system. The stages involved are indicated: macroplanning (Ma), microplanning (Mi) and/or surface realization (Re).

2007) “*Almost any shared resource will be good for the scientific advance of* NLG". Nevertheless, the author argues that the generation of quality resources for each of the specific stages, with its interfaces clearly identified, would contribute to a true advance in the field. In this sense, he makes the following proposal: that the researchers themselves make available to the community the resources used in their work with a suitable specification. Although at the same time, he assumes the difficulty of this task due to the cost associated with publishing and maintaining this type of product in the context of the research activity.

3.3 Relevant NLG Conferences and Workshops

Although NLG first appears in the 1950s as a minor part of machine translation, it was not until the 1980s when it became an important field of the NLP (McDonald, 2010). In this decade, the first workshops which focused their attention on NLG began to appear. The papers presented in these workshops usually solved small tasks related to the NLG field. During the 1990s, NLG research area grew considerably, with an increase in the number of conference papers presented, doctoral theses, workshops and research applications (Bernardos, 2007). Later, in the 2000s the first International Natural Language Generation Conference (INLG) was created, the most important conference in the NLG arena. Furthermore, the European Natural Language Generation Workshop (ENLG) was established replacing those workshops that emerged in the 1980s, becoming the other reference conference along with INLG. The INLG celebrated its 11th edition in 2018. The European conference, the ENLG, was held from 1987 until 2015. From 2015, following the celebration of the 9th INLG conference in 2016, it was decided that only the International one will be held in the future.

These conferences are the locus of the major advances in the NLG field as they provide a source of knowledge for those who want to delve into this research field. They have been held biannually but not together during the same year, that is to say, when the INLG was held, for example in 2014, the next ENLG would be held in 2015. The papers presented at these conferences were organized on the basis of which NLG task they resolved as well as the type or resource they provided. In addition, they had demo sessions where the performance of the systems shown, and, in recent editions of the INLG, tutorials and hackathons were organized.

Around these two conferences, different workshops have been organized on specific topics related to NLG. For instance, this is the case of the International Workshop on Natural Language Generation and the Semantic Web (WebNLG), whose goal is to promote the discussion and exchange of related research on NLG and the Semantic Web; or the Workshop on Computational Creativity in Natural Language Generation (CC-NLG) which is a workshop that brings together researchers from both communities, computational creativity and NLG.

Furthermore, apart from these conferences and workshops focused only on NLG, many papers related to the NLG field, are published in NLP general conferences. Among them, the most important are the ones organized by the Association for Computational Linguistics (ACL), including its international conferences and its North American Chapter of the Association for Computational Linguistics (NAACL) and its European Chapter of the Association for Computational Linguistics (EACL). Besides these, there are other ones to be considered as the Conference on Empirical Methods in Natural Language Processing (EMNLP), the International Conference on Computational Linguistics (COLING) or the International Conference on Text, Speech and Dialogue (TSD).

3.4 Conclusion

This third chapter presented an overview of the available resources concerning NLG. In this regard, the tools and corpus specific for NLG were described. Additionally, a review of the difficulty of evaluating the different parts of a NLG system along with the existing types of evaluation is provided. Also, information is outlined about the international conferences that are both specific for the NLG field and the ones for the NLP area in general.

With respect to the existing specific corpora, there are not as many as for other areas of NLP. This is because the corpora are usually created for specific applications, systems or concrete shared tasks. Subsequently, the format and the type of information stored in them differ from one to another, and their adaptation for use in other systems or approaches is costly. Regarding the tools mentioned in this chapter, some of them, such as SimpleNLG, have been employed independently in the development of other systems or approaches. The development of this type of tool allows the advancement of the research field since it presents solutions to complete NLG tasks.

Evaluating a NLG system is not an easy task, as this chapter has discussed. Although some metrics from other research areas can be used for evaluating the output of the system, they guarantee neither the quality of the generated text in terms of structure nor the meaningfulness. The lack of gold-standards in the evaluation of NLG systems due to the fact that there is not a singularly correct output for a specific input, makes a manual evaluation preferable in many cases. In this regard, it is common to perform the evaluation collaboratively with a concrete number of assessors. Therefore, this type of evaluation will be used in further Chapters to assess the results of our work. As aforementioned in the previous chapter, the interest in this research field has increased in the last years. This has been noted in the international conferences, where more and more specific workshops are being created for some of the tasks or topics in this arena. In addition to this, the main NLG conferences in conjunction with the top NLP ones are key to following the progress in this field.

Concerning the state-of-the-art in the NLG research area, their results cannot be measured as in other NLP areas. While the results in the task of automatic summarization are over 55% or in the task of postagging are over 99%, the resolution of the NLG task is complex to measure. This could be due to the fact that the task is not easy to evaluate automatically and the evaluation criteria are not well defined. The only results of the state-of-the-art that could be obtained are the ones from the challenges, when different systems are evaluated with the same corpus and under the same conditions. Knowledge on the diverse topics discussed in this chapter is essential when developing a new approach.

CHAPTER 4

HanaNLG: A Flexible Hybrid Approach for Natural Language Generation

As seen in the previous chapters (see Chapter 2 and Chapter 3), the process of NLG is composed of distinct and unique tasks that lead to the production of a text in natural language (Reiter & Dale, 2000). These tasks can be grouped into three different stages —macroplanning, microplanning and surface realization—. Surface realization is responsible for creating the final output of a NLG system, whether in text or speech form. This work seeks to validate the hypothesis that the use of hybrid techniques during the development of the tasks included in a NLG system will allow the generation of high quality texts in a flexible way, that is aims to be independent of domain and language. This thesis will exclusively address one of the stages previously mentioned, which is, in this case, the surface realization stage from a hybrid perspective.

From this perspective, Chapter 4 provides our proposal for tackling the surface realization stage. HanaNLG (Hybrid surfAce realisatioN Approach for Natural Language Generation) is a hybrid approach for the surface realization stage, capable of automatically generating text that is easily adaptable to different genres, domains and languages. HanaNLG is hybrid because it relies on the use of linguistic resources as well as on statistical information, through the use of FLM (see Section 2.3.2 of Chapter 2), to construct the final output. The use of these types of resources along the generation approach results in the increase in the naturalness of the language as well as its quality. In addition, since HanaNLG is only focused on the surface realization stage and we do not have the macroplanning process, in order to guide a generation based on a certain theme, words, domain, etc. we propose the seed feature concept. These seed features can be seen as abstract objects (e.g. phonemes, sentiments, polarities, etc.) which will guide the generation process in terms of vocabulary. Consequently, the type of texts generated by HanaNLG can be adapted to different domains and also for different communicative goals (e.g. summarization). Moreover, given the nature of the resources and techniques employed, our approach is also easily adaptable to different genres, domains and languages. Hence, our contributions to the field are as follows: (i) a novel hybrid approach combining statistical information with linguistic resources; (ii) the flexibility to be able to adapt the generation of text for different domains, purposes and languages; and, (iii) the variety of language to appear in the generated text is increased. Our work primarily differs from the existing NLG systems because the generation process has the flexibility to adapt to different purposes, domains and languages.

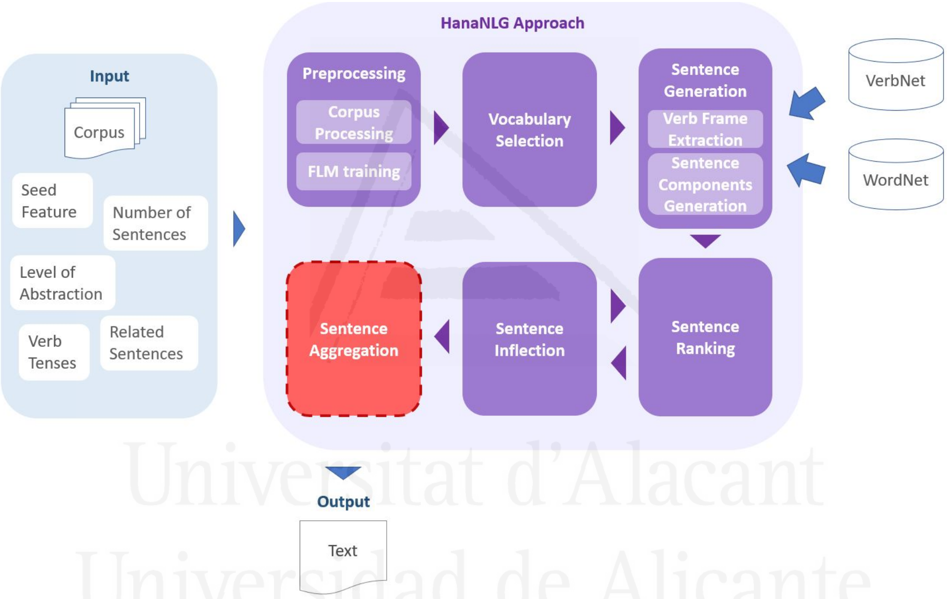


Figure 4.1: Architecture of HanaNLG.

HanaNLG is structured in an architecture of six distinct modules. An overall scheme of this architecture can be seen in Figure 4.1. The inputs to the system are: i) a corpus; ii) a seed feature; iii) the number of sentences; iv) level of abstraction (i.e. using words, lemmas or synsets[1] for the generation of the sentence); and, optionally v) a boolean variable indicating the generation of related sentences (i.e. sentences with their subject or object related) and/or vi) sentence verb tenses (i.e., this one will be only used during the *Sentence Inflection* module if it is supplied, otherwise see Section 4.5). The input corpus will be used to train the language models employed during the generation as well as to gather information about words in the *Vocabulary Selection* module. The final output of HanaNLG is composed of sentences in natural language whose content is easily adapted to

[1] Set of cognitive synonyms related to a term or concept used in WordNet.

the desired domain they have been generated for. Thus, we want to validate the hypothesis that the use of hybrid techniques will result in a higher quality text. A brief description of these modules, which will be further detailed in the next sections, is as follows:

- *Preprocessing*: This module is in charge of processing the input data as well as the input corpus to be used during the generation process. Depending on the application for which the text is going to be generated, the input to HanaNLG may need some processing in order to adapt this data to the generation process. For instance, if the documents of the input corpus are plain texts (i.e. they do not have format or are not tagged), it will need to be preprocessed to be used for training the language models.

- *Vocabulary Selection*: Once the input data has been preprocessed, HanaNLG selects the words that will form the vocabulary based on the seed feature introduced as input. This vocabulary will be preferably employed during the generation.

- *Sentence Generation*: Taking as input the vocabulary from the previous module, this module is responsible for generating sentences following an over-generation strategy. These sentences will be generated using linguistic resources as well as statistical information.

- *Sentence Ranking*: This module will select one sentence, among the ones generated in the *Sentence Generation* module, which will form part of the final output of HanaNLG. This ranking is performed employing FLM.

- *Sentence Inflection*: When a sentence is selected by the previous module, its words are inflected employing information from lexicon resources. This inflection is necessary since the words in the sentence may not be concordant in number or gender, or they may be in lemma or in synset form, making their transformation into words mandatory.

- *Sentence Aggregation*: Once all the sentences are generated, this module is in charge of avoiding the repetition of sentences and the redundancy. This last module is optional, being performed only when more than two sentences are generated.

Although these modules do not follow the tasks strictly defined for the surface realization stage, they were developed in this way to ensure that the generation process was as flexible and adaptable as possible. However, due to the fact that some of the linguistic resources used in the development of HanaNLG are only in English, the current proposal of HanaNLG only generates texts in English. In the case that those linguistic resources were released in other languages, HanaNLG will be capable of generating text in those languages.

The whole process is capable of producing a desired number of sentences, which is provided as one of the inputs of HanaNLG as can be seen in Figure 4.1. However, the process of generating one or several sentences is the same. Therefore, this chapter will provide a detailed description of the process to generate a sentence in the following sections, which will be focused on the modules of HanaNLG. The first module of HanaNLG, where the input data is processed in order to be used during the generation process, is described in Section 4.1. Once the input data is processed, the words to be used for the generation will be chosen as described in Section 4.2. Next, a set of sentences will be generated using over-generation techniques (Section 4.3), and one of them will be selected to form part of the output of HanaNLG (Section 4.4). Then, the words within the selected sentence will be inflected as described in Section 4.5. Once all the desired number of sentences are generated, they are analyzed and aggregated if necessary as detailed in Section 4.6. Finally, the last section (Section 4.8) concludes this chapter analyzing the main contributions. In each of these stages, an illustrative example of how they work will be shown.

4.1 Preprocessing

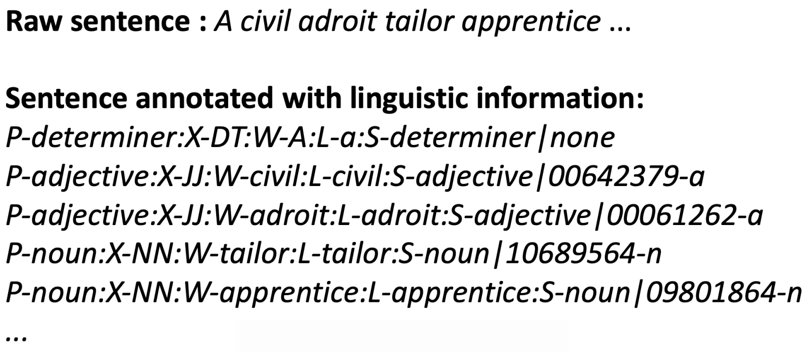
This section describes the preprocessing module of HanaNLG, which is the first one in the architecture and can be seen in Figure 4.1.

The main objective of this module is to process the input of HanaNLG (i.e. a corpus, a seed feature, the number of sentences, the level of abstraction and, optionally, a variable indicating the generation of related sentences and the verb tenses of the sentences) so that it can be used in the generation process. Specifically, since the input corpus is usually raw text documents, this is the only element of the input that needs to be processed. This corpus will be used for different tasks within HanaNLG: i) training the FLM used in the generation of sentences and the ranking; and, ii) gathering word information during the generation process. Therefore, the description of the corpus processing along with the training of the FLM will be detailed in the following subsections.

4.1.1 Corpus Processing

The first step before starting the generation of a sentence in HanaNLG is to analyze the input data as well as to adapt its information to be usable. In this case, the main information source, which will determine the basis for the content of the final text, comes from the input corpus.

This input corpus consists of a set of text documents, which have neither a format nor have they been tagged. Thus, it is necessary to first perform a linguistic analysis to gather different types of information of the words comprising the input corpus. Specifically, information about the words themselves, their lemma and POS tag, and also about their synset is obtained. In this respect, the language analyzer Freeling (Padró & Stanilovsky, 2012) is used to perform a linguistic analysis at different levels (e.g. lexical, syntactic and semantic). This information is used to automatically tag all the input corpus in the corpus preprocessing task, resulting in a text similar to the one shown in Figure 4.2.



Raw sentence : *A civil adroit tailor apprentice …*

Sentence annotated with linguistic information:
P-determiner:X-DT:W-A:L-a:S-determiner|none
P-adjective:X-JJ:W-civil:L-civil:S-adjective|00642379-a
P-adjective:X-JJ:W-adroit:L-adroit:S-adjective|00061262-a
P-noun:X-NN:W-tailor:L-tailor:S-noun|10689564-n
P-noun:X-NN:W-apprentice:L-apprentice:S-noun|09801864-n
…

Figure 4.2: Example of the format of the corpus, where *P*: simple POS tag; *X*: full POS tag; *W*: word; *L*: lemma; and *S*: simple POS tag+synset. The data used for tagging the corpus is obtained from the linguistic analysis performed by the tool Freeling.

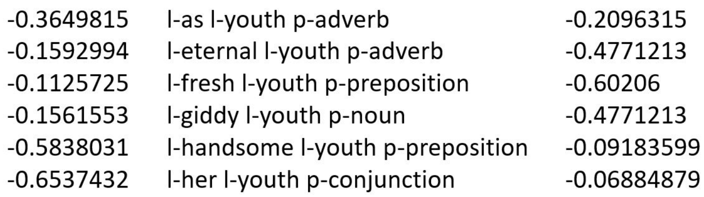
The tagged text in the format shown in Figure 4.2 is the one suitable for training the FLM which will be described in the next subsection.

4.1.2 Factored Language Models Training

Once the corpus is tagged, the next step before starting the generation process is to train the FLM over it. As previously mentioned, these language models will be used in several modules of HanaNLG in different ways. For example, some of the trained models will be used for generating the sentences and others only for computing the sentence probability (see Section 4.4).

In Section 2.3.2 of Chapter 2, it was established that the two key issues to be taken into account when training FLM were: (i) to choose an appropriate set of factors; and, (ii) to find the best probabilistic models over these factors. In this regard, the information which was automatically tagged in the corpus (i.e. the words, their lemma, their POS tag and their synset) is used as the factors for training the different FLM. The reason behind the selection of these factors was that they can provide more flexibility to the generated text in terms of vocabulary. This is because the words within the synsets, which have the same semantic meaning, can be exchanged depending on the context. For example, the word *cat* or the word *kitten* would be used in a children's story while the word *felis catus* would be used in a more technical context.

Regarding the type of FLM used, the trigram probabilistic model was selected due to its simplicity and usability in the NLP area. An example of a trigram model that would be trained in this task is shown in Figure 4.3.



-0.3649815	l-as l-youth p-adverb	-0.2096315
-0.1592994	l-eternal l-youth p-adverb	-0.4771213
-0.1125725	l-fresh l-youth p-preposition	-0.60206
-0.1561553	l-giddy l-youth p-noun	-0.4771213
-0.5838031	l-handsome l-youth p-preposition	-0.09183599
-0.6537432	l-her l-youth p-conjunction	-0.06884879

Figure 4.3: Example of a trigram FLM whose factors are lemmas (l-) and POS tags (p-).

4.2 Vocabulary Selection

Before starting the process of generating a sentence, it is essential to have information about the content that it will contain. In the case of having a macroplanning stage, this module would take into account the information provided by the macroplanning. However, since in this work we only focus on the surface realization stage, the vocabulary obtained in this module will be based on the seed feature given as input. This module corresponds to the second one in the architecture of HanaNLG (see Figure 4.1).

A seed feature can be considered as an abstract object (e.g. a word, a phoneme, a sentiment, etc.) that guides the generation process in terms of vocabulary; hence, its importance within HanaNLG. In this regard, HanaNLG has been applied to different contexts where different types of seed features where used, that will be discussed in Chapter 5. Each one of these seed features has their own distinctive characteristics. For instance, we can consider phonemes as the seed feature to generate a sentence, that would be useful, for example, in phonetic therapies. In the case of considering polarity as the seed feature, a sentence with positive and negative polarity could be generated to support users or systems with the generation of reviews and evaluative text. If we contemplate a keyword as a seed feature, sentences including those specific words can be generated. The keyword seed feature may be useful when it is necessary to generate sentences with specific information, such as summaries or headlines. This seed feature could also be the characters of a story, and the story can be generated based on these characters and their actions. Alternatively, the seed feature could also be a sentiment, where sentences could be produced reflecting sentiments, such as happiness or fear. Some of these seed features will be described in more detail in Chapters 5 and 6.

The detection of these types of words is complex. Some of them may require a linguistic resource, such as lexicons or corpus. For example, in order to detect words with a specific phoneme, we could use the specific grapheme for each phoneme. Or, in the case of the keywords, the words to be detected may be already provided or they could have been obtained from a keyword/topic detection system. Figure 4.4 shows the words that this module may select for a specific seed feature in the context of generating sentences for phonetic therapies.

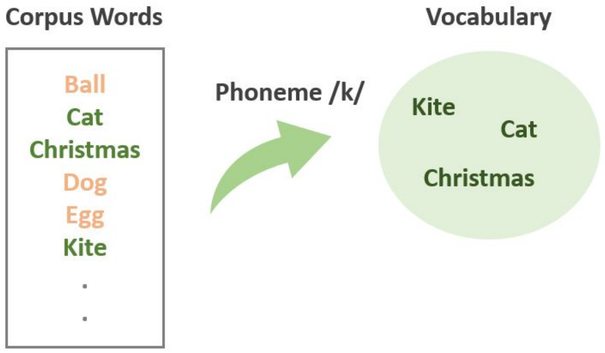


Figure 4.4: Example of the selected words from the corpus for the phoneme /k/.

Once the words for a specific seed feature are identified, they are stored in a bag of words. This bag of words will be used during the generation process to ideally select the words contained in it to be included in the generated sentence.

4.3 Sentence Generation

This section delineates the process of generating a sentence based on over-generation techniques [2]. The aim of this module is to generate a sentence containing and maximizing the vocabulary related with the seed feature and which is obtained in the previous module. The inputs to this module are the vocabulary selected in the previous module (see Section 4.2), the FLM previously trained (see Section 4.1.2) and the *level of abstraction* given as input to the approach. The *level of abstraction* is a variable which indicates what type of information will be used to generate the sentence. This type of information can range from more concrete elements, such as words (e.g. “cat" or “man"), to more complex and abstract ones, as in the case of synsets (e.g. “02985606-n" or “10287213-n"). In this sense, HanaNLG can handle three different *levels of abstractions*: i) the word as is; ii) the word in lemma[3] form; or, iii) the synset of the word. This type of element, i.e., the *level of abstraction*, increases the flexibility of the approach since the lemmas can be put in feminine or plural and the synsets can correspond to a certain number of words with the same meaning.

The output of this module is a set of sentences that will be ranked in the next module (see Section 4.4). This module corresponds to the third one in the architecture described in Figure 4.1.

HanaNLG generates the sentence from its core, which in this case is the verb of the sentence. Then, the rest of the sentence is produced based on the

[2] In these types of techniques, several sentences are generated and then ranked, based on their probability, in order to only select the one with the highest probability.

[3] A lemma is the canonical form or the dictionary form of a set of words.

characteristics of this verb. In order to do this, the process makes use of linguistic resources as well as statistical information, following two distinct steps.

- *Verb Frame Extraction*: Starting from the vocabulary, several verbs are selected and their frames are extracted. These frames comprise the structure of the sentence.
- *Sentence Components Generation*: Once the structure of the sentence is known, its components are generated, resulting in a complete sentence.

4.3.1 Verb Frame Extraction

As mentioned before, the sentences in HanaNLG are generated starting with the verb. The verb is the part of the sentence that expresses actions, movements, conditions, etc;. In traditional grammar, it can be conceived as the main word of the predicate in a sentence. Consequently, in this approach, the verb is considered to be the core of the generation process. Therefore, the sentences will be generated using linguistic information related to the verb.

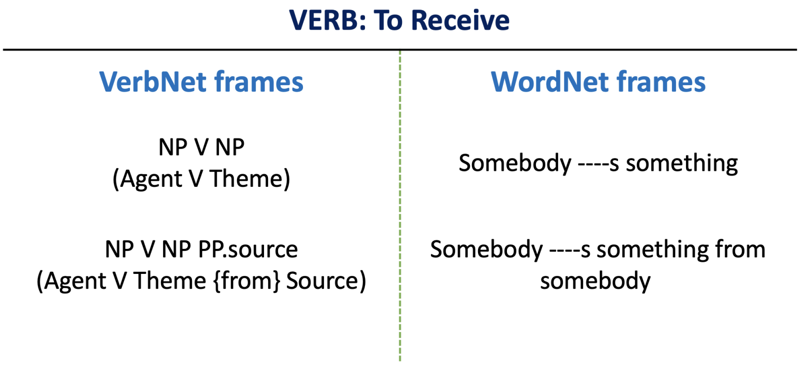
For this purpose, lexical resources are employed to obtain syntactic information associated with the verb. Specifically, the lexical resources VerbNet (Schuler, 2005) and WordNet are used to obtain syntactic frames to generate the sentences. On the one hand, VerbNet is one of the largest verb lexicon for English and contains syntactic information as well as semantic information about verbs. On the other hand, WordNet is a lexical database whose elements are grouped into synsets. The frames gathered from VerbNet contain different information than the ones from WordNet. In the case of the frames from VerbNet, they contain both syntactic and semantic information about the verbs included in the lexicon. However, the frames from WordNet are a set of generic frames for all the verbs and only provide simple syntactic information.

In order to generate several sentences, a set of verbs is collected from the bag of words (i.e. the vocabulary related to the input seed feature). Both the frames from VerbNet and WordNet are extracted for each of these verbs. These frames will be used in the next step of the module to generate a sentence for each of them. In the case that there is no verb within the vocabulary, a set of the most common verbs[4] in the input corpus is collected. Then, the frames will be extracted and passed to the next step of the process for the generation of the sentences. Figure 4.5 shows the frames that this module would extract for the verb *to receive*.

4.3.2 Sentence Components Generation

Once all the frames of the verbs from the vocabulary are extracted, a sentence for each of them is generated. In order to produce this sentence, the main elements of the sentence need to be generated. In this regard, taking into account the

[4] In the selection of common verbs in the corpus, the modal and auxiliary verbs are excluded since they do not provide enough useful information to form a sentence.



VERB: To Receive

VerbNet frames	WordNet frames
NP V NP (Agent V Theme)	Somebody ----s something
NP V NP PP.source (Agent V Theme {from} Source)	Somebody ----s something from somebody

Figure 4.5: Example of the frames obtained from VerbNet and WordNet for the verb *to receive*.

information given by the verb frame, an analysis is performed to which of the constituents of the sentence need to be generated. For this case, if the frame denotes that the verb needs a *Subject*, this will be generated first. Likewise, if the *Object* of the sentence is needed, then this will be generated afterwards. These two elements are usually considered to be the most important in a sentence given that one of them is talking about *who* is performing an action, and the other one refers to the *person, object, etc.* that is receiving that action (Carter & McCarthy, 2006). All of these elements are generated by using the same process, independently of the *level of abstraction* given as input. This process employs the trained FLM to search for the element needed (i.e. the element needed could be different things such as a person, an actor, an instrument, an animal, a preposition, etc; and WordNet is used for recognizing the words that fit with the mentioned categories) with the highest probability appearing with the verb core (whether it is the word, the lemma or the synset) always prioritizing the words in the vocabulary. For instance, if we consider the scenario in Figure 4.6, where we are generating sentences for the phoneme /oo/, the verb core is "to carve", and the VerbNet's frame for generating the sentence is "NP.material V NP", the first step would be to select the word for generating the subject of the sentence. In this case, based on the characteristics of the frame, the word of this subject needs to be a *material*. Therefore, we first look into the vocabulary for words that are *material*. In the figure can be seen that there are two different words that fit in this category (i.e., wood and wool). The second step would be to search in the FLM if any of these words appear together with the verb "to carve", which, in this case, as seen in the figure the word "wood" appear. Therefore, this word (i.e., wood) would be the one chosen to be the subject of the sentence. In the case that the word "wool" also appeared in the FLM, the words selected would be the one with the highest probability.

As mentioned before, the generated sentences can be related in terms of their elements. In these cases, the frame is first analyzed to verify if any of its

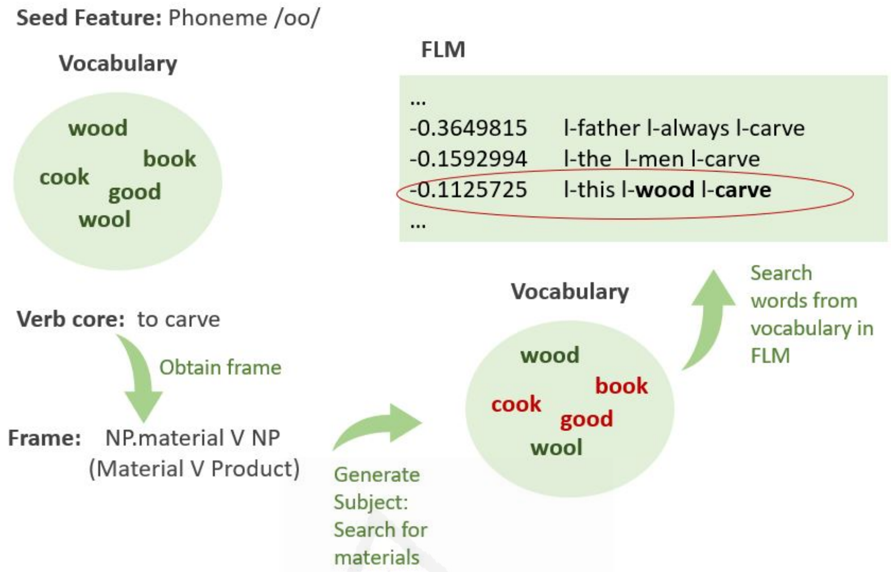

Seed Feature: Phoneme /oo/
Vocabulary
wood
book
cook
good
wool
Verb core: to carve
Obtain frame
Frame: NP.material V NP
(Material V Product)
Generate Subject: Search for materials
Vocabulary
wood
book
cook
good
wool
Search words from vocabulary in FLM
FLM
...
-0.3649815 l-father l-always l-carve
-0.1592994 l-the l-men l-carve
-0.1125725 l-this l-wood l-carve
...

Figure 4.6: Illustrative example of how HanaNLG performs the selection of the subject's word for a specific frame given the verb core "to carve". In this case, the frame indicates that the subject need to be a *material*, therefore, HanaNLG looks for a word in the vocabulary that matches this characteristics and that also appears in the FLM with the verb core.

elements (i.e. whether the subject or the object) matches any of the elements of the previously generated sentence[5]. The elements need to match in terms of type of elements and type of information required. For example, if the previous selected subject is an animal but the subject of the current sentence needs to be a human then, they do not match in type. Otherwise, that element will form part of the current sentence, being the current sentence and the previous one related to that element.

Once the elements are generated separately, they are arranged to form the sentence for that specific frame. Depending on the *level of abstraction* employed, the sentences will have different aspect. Example 8 shows several sentences generated by this module in their word, lemma and synset form from the frames in Figure 4.5.

(8) **Word**
NP V NP → Mary receives a cat.
NP V NP PP.source → Mary received a kite from Sara.
Somebody —-s something → Mary received a kite.
Somebody —-s something from somebody → Mary receives a cat from her mum.

Lemma

[5] When HanaNLG is generating the first sentence of the output, this is not checked.

NP V NP → mary receive a cat.
NP V NP PP.source → mary receive a kite from sara.
Somebody —-s something → mary receive a kite.
Somebody —-s something from somebody → mary receive a cat from her mum.

Synset
NP V NP → noun|11161412-n verb|02210119-v determiner|none noun|02121620-n punctuation|none

NP V NP PP.source → noun|11161412-n verb|02210119-v determiner|none noun|03621473-n preposition|none noun|none punctuation|none

Somebody —-s something → noun|11161412-n verb|02210119-v determiner|none noun|03621473-n punctuation|none

Somebody —-s something from somebody → noun|11161412-n verb|02210119-v determiner|none noun|02121620-n preposition|none pronoun|none noun|04652345-n punctuation|none

4.4 Sentence Ranking

This section explains the ranking methodology followed in HanaNLG, which determines the sentences that will form part of the final output of HanaNLG. This module corresponds to the fourth one depicted in the architecture (see Figure 4.1).

Since this approach is following an over-generation and ranking strategy, the next step, after generating a set of sentences, is to perform a ranking. This ranking will select the most appropriate sentence from all the ones generated in the previous module.

In order to do this, the sentences are ranked based on their probability. The probability of a sentence is computed following the chain rule, which can be seen in Equation (4.1).

$$P(w_1, w_2 ... w_n) = \prod_{i=1}^{n} P(w_i | w_1, w_2 ... w_{i-1}) \tag{4.1}$$

Based on the chain rule, the probability of a sentence can be calculated as the product of the probabilities of their words. Depending on the language model employed the probability of a word can be calculated in different ways. In this regard, the FLM language model is used for computing the probability of the sentences. This probability is calculated as a linear combination of FLM as suggested in (Isard et al., 2006). In this linear combination a weight λ_i is assigned to each of the FLM used, being 1 their total sum. In Equation (4.2) this linear combination can be seen, where f refers to the factors selected for the different FLM employed.

$$P(f_i | f_{i-2}^{i-1}) = \lambda_1 P_1(f_i | f_{i-2}^{i-1})^{1/n} + \cdots + \lambda_n P_n(f_i | f_{i-2}^{i-1})^{1/n} \tag{4.2}$$

The sentence that would form part of the final output of HanaNLG will be the one with the highest probability besides having the maximum number of words with the selected seed feature. In Example 9 the final sentence, which this module would select from the ones in lemma form from Example 8, is shown. In the case of generating the sentences with words or synsets, the procedure is the same as in the case of lemmas. The probability would be calculated based on the FLM trained with words or synsets as a factor.

(9) **Generated Sentences**
mary receive a cat. → Probability: 0.25
mary receive a kite from sara. → Probability: 0.03
mary receive a kite. → Probability: 0.1
mary receive a cat from her mum. → Probability: 0.08

Selected Sentence
mary receive a cat.

When a sentence is chosen, the next step before adding it to the final output is to inflect the words within the sentence which will be detailed in the following section.

4.5 Sentence Inflection

Morphological inflection is key when we are talking about natural language. Without it, the information within a sentence cannot be correctly understood, since we lose the reference of time and the person who is performing the action. Thus, this step is indispensable to make the language more natural and fluent. The main objective of this section is to detail the inflection process carried out by HanaNLG, which is the last module of the architecture of this approach.

On the basis of the definition of inflection[6]: “*the change of form that words undergo to mark such distinctions as those of case, gender, number, tense, person, mood, or voice*"; this is a characteristic common to many languages that allows the concordance of gender and number. For instance, in English, we can transform the singular word “dog" to its plural form “dogs" by adding an “-s" at the end of the word. In Spanish, the word “*chico*" (boy) that is masculine, could be converted to feminine just by exchanging the final “-o" for an “-a", resulting in the word “*chica*" (girl). Or more complex inflections related to verbs, such as the several changes in the Spanish verb “*elegir*" (to choose) to transform it into the first person singular of the present of the subjunctive, resulting in the word “*elija*" (choose).

The rules required to inflect words vary from one language to another. Therefore, this module needs to be trained or adapted depending on the target language. In the next chapter (Chapter 5), the adaptation of this module to different languages will be described.

[6] https://www.merriam-webster.com/dictionary/inflection

Regardless of the language employed, this module will make minor generic changes to the sentences depending on the level of abstraction (i.e., word, lemma or synset) provided as input of HanaNLG. In the case that the sentence was generated using words, the changes to the words in the sentence may only affect the concordance with singularity and plurality and grammatical person (first person, third person singular or others). These changes will be made based on the verb characteristics (i.e. tense, person and number). Example 10 shows the changes that this module would made to a sentence in English with these characteristics.

(10) The girls plays the piano. → The girl plays the piano.

In the case that the sentence was generated using lemmas or synsets, there are several issues to decide. Regarding the lemmas, the module already has the word to inflect, but with the synsets we only have the identification of WordNet for the set of synonyms. Therefore, it is essential to choose the final words that will form the sentence in order to be able to inflect the sentence. Since a synset may be related to several words, each synset of the sentence is expanded into all the synonyms. These synonyms are usually in lemma form. Then a set of sentences, with their words in lemma, is generated with all the possible combinations of the synonyms. An example of how this module would perform the expansion of a sentence with their words in synset, for the English language, is shown in Example 11.

(11) **Generated sentence in synset form**
noun|11161412-n verb|02210119-v determiner|none noun|02121620-n punctuation|none

Expanded sentences
Mary receives a cat.
Mary receives a true cat.
Mary receives a domestic cat.
Mary receives a feline.
Mary received a cat.
Mary has a cat.
Mary had a cat.
...

Once the sentence generated with synsets has been transformed into a set of sentences whose words are in lemma form, the rest of the inflection process is the same for the lemma and synset case. This process starts with the verb inflection which has two distinct configurations. On the one hand, the verb tense can be configured beforehand, allowing the user to select the desired verb tense for each sentence. On the other hand, the verb tense will be automatically selected. In this case, this module will generate an inflected sentence for each target language verb tense, grammatical person, singularity and plurality. Then, the remaining words in these sentences are inflected following the characteristics of the selected verb, as in the case that the sentence from the previous stage was generated using words.

When the verb tense is not defined or the sentence is generated using synsets, several candidate inflected sentences are generated as previously mentioned. Therefore, a ranking is necessary to only select the most appropriate candidate sentence based on the probability and the seed feature. This ranking is conducted by the *Sentence Ranking* module described in the previous section. In Example 12 is shown the sentence, from the ones in Example 11, which will be finally selected by the *Sentence Ranking* module. In the case of a draw, as seen in this Example, the selected sentence would be the first one of them.

(12) **Inflected Sentences**
Mary receives a cat. → Probability: 0.15
Mary receives a true cat. → Probability: 0.001
Mary receives a domestic cat. → Probability: 0.02
Mary receives a feline. → Probability: 0.003
Mary received a cat. → Probability: 0.2
Mary has a cat. → Probability: 0.15
Mary had a cat. → Probability: 0.05
...

Selected Sentence
Mary received a cat.

The output of this module is an inflected sentence in natural language which will form part of the final output of HanaNLG.

4.6 Sentence Aggregation

This section describes the process of sentence aggregation once all the sentences are generated by HanaNLG. This last module is optional and it would only be performed when more than one sentence is generated. This type of aggregation is necessary at the end of the generation process to avoid the repetition within the output, as well as information redundancy (Dalianis, 1999).

In this regard, it is possible that the same subject is generated several times in consecutive sentences due to the generation of related sentences. Therefore, the information provided by these sentences may be merged into only one sentence. This step is usually performed in the microplanning stage of the NLG pipeline. However, since we are only addressing the surface realization we also integrate this aspect of the microplanning stage within our approach.

For this purpose, we make a rule-based aggregation. Concretely, at this moment, the aggregation performed only affects the subject and the verb of the sentences. Therefore, two types of rules are employed:

- **Rule 1**: Two consecutive sentences are merged when their subjects and verbs coincide. Example:
  “Mary is young. Mary is nice."
  “Mary is young and nice."

- **Rule 2**: Two consecutive sentences are merged when their subjects coincide but not the verb. Example:
  “Mary is young. Mary travels a lot."
  "Mary is young and travels a lot."

These rules are quite similar but serve the purpose of avoiding the redundancy of the final output of HanaNLG. Example 13 shows how this module would carry out the aggregation on the sentences in the Example 12 to obtain the final output of this approach.

(13) **Output of HanaNLG**
A family was celebrating Christmas.
Sara gave her children some presents.
Mary received a cat.
Mary received a kite.
Sara cooked the dinner.
Sara called her sister.
...

Aggregated output
A family was celebrating Christmas.
Sara gave her children some presents.
Mary received a cat and a kite.
Sara cooked the dinner and called her sister.
...

4.7 Tools

During the construction of HanaNLG some tools were used for handling some types of data or for training the language models. In this regard, as previously mentioned in Section 4.1, the language analyzer tool Freeling was used to perform a linguistic analysis at different levels to the input corpus. This tool can work with many languages, including English and Spanish. In order to train the language models, the SRILM software (Stolcke, 2002) was employed. This software allows the building and training of different language models and includes an implementation of the FLM. For handling WordNet, the library JWI (Finlayson, 2014) was used, and in the case of VerbNet, the library JVerbnet[7] was employed instead. Table 4.1 shows in which module are these tools employed.

4.8 Conclusion

This chapter presented HanaNLG, a hybrid approach to address the surface realization stage which can generate natural language. This approach can generate one or more sentences following an over-generation and ranking strategy. Our approach is prepared to be adapted to different domains and types of texts. In

[7] http://projects.csail.mit.edu/jverbnet/

Tool	Module	Usage
Freeling (Padró & Stanilovsky, 2012)	Preprocessing	Preprocess the corpus in order to gather lexical, syntactic and semantic information about words.
SRILM (Stolcke, 2002)	Preprocessing	Train and build the FLM used during the sentence generation and sentence ranking modules.
JWI (Finlayson, 2014)	Sentence Generation, Sentence Inflection	Handle the information provided by WordNet.
JVerbnet[7]	Sentence Generation	Handle the information provided by VerbNet.

Table 4.1: Usage of the tools in the different modules of HanaNLG

order to perform this adaptation, HanaNLG makes use of input seed features to guide the generation in terms of the vocabulary. With the purpose of producing such types of texts, an architecture comprising six modules is proposed. First, a preprocessing is required to adapt the input data to be usable during the generation process (*Preprocessing*). Next, the vocabulary to be used in this process is selected (*Vocabulary Selection*). With this vocabulary, a set of sentences is generated (*Sentence Generation*) and subsequently ranked (Sentence Ranking). Then, when a sentence is finally selected to be included in the output of HanaNLG, the sentence is inflected to make the language as natural as possible (*Sentence Inflection*). Finally, when all the text is generated it is analyzed in order to avoid repetition and redundancy (*Sentence Aggregation*).

Regarding the advantages of HanaNLG, it is worth mentioning that given this hybrid approach, the combination of statistical information with linguistic resources provides flexibility to HanaNLG. Moreover, flexibility with respect to the generation of text provides adaptability to diverse domains. Additionally, the use of different levels of abstraction, given as input to HanaNLG, contributes to this flexibility in terms of the variability of vocabulary. The initial hypothesis is that the combination of statistical and knowledge-based approaches was beneficial for NLG because it potentially results in higher quality generated language. As long as the language-dependent linguistic resources (WordNet, Verbnet, ...) were available for the target language, HanaNLG would be easily adaptable to a multilingual environment. To recapitulate, the main contributions of HanaNLG with respect to the state of the art of the NLG are:

To recapitulate, the main contributions of HanaNLG with respect to the state of the art of the NLG are:

- **The proposal of a novel hybrid approach for the surface realization stage**. This approach combines the use of statistical language models (i.e., FLM) and linguistics resources (i.e., VerbNet and WordNet) for generating language regardless of the domain and purpose.

- **The flexibility to generate natural language that can be easily adapted to different domains and purposes**. Thanks to the use of seed features, the text can be adapted for a specified domain. At this moment, several types of seed features are implemented which will be described in the next chapter (Chapter 5).

- **Diversity of vocabulary**. The use of a different level of abstraction during the generation process allows access to a wide range of options when inflecting a sentence. Specifically, in HanaNLG two distinct levels of abstraction permit this: the lemma and the synset.

The following chapters will validate the hypothesis that the language generated by HanaNLG is natural and of good quality.

CHAPTER 5

HanaNLG Intrinsic Evaluation

The previous chapter described HanaNLG, the proposed approach to address the surface realization stage from a hybrid perspective. A robust experimentation and evaluation are necessary in order to verify the effectiveness of HanaNLG. Therefore, the main objective of this chapter is to conduct an intrinsic evaluation of HanaNLG.

For this evaluation, we will focus on the assessment of the text generated by HanaNLG in terms of different aspects such as, for example, the structure of the text, its content or its meaningfulness. Since the evaluation of a NLG is a complex task as mentioned in Chapter 3, we cannot only evaluate the generated text automatically. This is due to the fact that there is not a single correct output to the system and neither are gold-standards to compare the output, in contrast to other NLP fields. Besides that, existing metrics, which do not belong to the NLG scope, are unable to automatically measure the coherence, the meaning and the structure of a given text. Subsequently, in almost all the cases, we perform a manual and collaborative evaluation with assessors to evaluate the output of HanaNLG.

To facilitate further analysis of HanaNLG's performance, this chapter focuses on the following questions that were raised during the design and development of the various modules of the proposed approach:

- What type of LM is most appropriate for generating language in our architecture?
- Does the use of seed features in conjunction with FLM allow the generation of language for different domains?
- Does the combination of semantic and syntactic information benefit the quality of the text generated?

This section will also describe the environment in which the aspects of

HanaNLG has been evaluated. This includes the description of the seed features tested, the configuration of the inflection module for different languages, or the scenarios proposed to assess the flexibility of HanaNLG.

The experimentation and the evaluation of HanaNLG will be addressed as follows in the rest of the chapter. Section 5.1 outlines the experimentation setup, including the different types of seed features implemented (Section 5.1.1), the different configurations of the inflection module (Section 5.1.2) and the scenarios tested (Section 5.1.3). In Section 5.2 some research questions, that will define the aspects to assess, are laid out. Specifically, we proposed 6 evaluation questions that will be answered from Section 5.3 to Section 5.8. Finally, conclusions and some insights about the future work are discussed in Section 5.9.

5.1 Experimentation Environment

In order to assess the performance of HanaNLG several issues have to be taken into account. Firstly, concerning the internal configuration and development of our approach, there are multiple variables that can be configured. Secondly, issues that affect the experimentation setup, such as the scenarios wherein the experimentation was focused.

In this sense, within HanaNLG, the type of seed feature introduced as input must be adjustable to permit flexibility so that the sentence generated is adaptable to different purposes. Specifically, there are diverse types of seed features already implemented and, in addition to this, more seed features can be introduced implying a relatively low cost without affecting the structure of our approach. Moreover, the inflection module within HanaNLG can handle different languages: English and Spanish. Regarding the experimentation itself, different scenarios are proposed. All these issues are as follows.

5.1.1 Types of Seed Features

Seed features are a key component of HanaNLG. They provide us with a means by which to easily adapt the generation process to diverse domains, targets or purposes. In this respect, some seed features were evaluated, but our approach is constructed in such a way that the inclusion of new features minimizes cost without affecting the approach itself. This will allow more functionalities and purposes to be added to the approach in the future.

The assessed seed features are: i) phonemes, ii) polarity and iii) keywords or important words. As aforementioned in Section 4.2 of Chapter 4, these seed features could be useful in the generation of text for different contexts.

In the context of assistive technologies, phonemes could be employed in the generation of sentences for specific speech therapies. The detection of this kind of seed feature is not easy because it depends on the target language. During this experimentation, we worked with two languages, i.e. English and Spanish, and therefore, the resources used in the detection of phonemes differ from one language to another. In both cases, the phonemes are detected by their graphemes[1]. In the case of English, the phonemes are detected by employing the graphemes provided by *the Reading Well* [2], whereas for Spanish the phonetic restrictions exposed in (Morales, 1992) are used.

In the context of constructing new resources, the second seed feature evaluated could be useful. In this sense, the sentence generated with polarity may be employed for helping in the creation of resources such as corpora, tagged datasets, etc. In this case, its detection in a corpus is easier compared to the detection of phonemes. This is due to the fact that there are many datasets and lexicons containing words with a positive and negative polarity. Concerning the present work, the words that are related to a specific polarity from (B. Liu et al., 2005) and the ML-SentiCon (Cruz et al., 2014) lexicon were used to detect the words related to this type of seed feature.

Finally, in the context of topic-focused texts, taking keywords or important words as seed features could be useful in the creation of texts related to a certain theme, such as summaries or news headlines. These seed features may also help in the production of other types of texts, such as stories. In order to work with such seed features, as list of keywords or important words may be provided beforehand, or they could be obtained, for example, through the use of a topic detection tool or through the macroplanning stage in a complete NLG system.

5.1.2 Inflection configuration

As seen in the previous chapter, morphological inflection is an essential part of the natural language. The inflection module within HanaNLG is designed to work regardless of the language employed; however, the selection of the inflected form of a specific word is language-dependent. In this sense, this module can handle the inflection of English and Spanish sentences.

English Inflection

Regarding the inflection for the English language, this is addressed by the use of lexicons. Specifically, the lexicons included in Freeling (Padró & Stanilovsky, 2012) are used for English. This lexicon contains about 68,000 forms (including information about words, lemmas, POS tags or synsets), which were automatically extracted from the Wall Street Journal corpus (Charniak & al., 2000).

This lexicon is used to obtain the desired word form based on the verb characteristics. In this sense, in Chapter 4 we explained that HanaNLG can handle different levels of abstraction (i.e. words, lemmas and synsets) when generating a sentence. Therefore, when a sentence is generated with words, in order to inflect a specific word within the sentence, we first linguistically analyze this

[1] A grapheme is the set of letters or combination of letters that represent a phoneme.

[2] http://www.dyslexia-reading-well.com/support-files/the-44-phonemes-of-english.pdf

word to obtain its lemma. Then, we use that lemma to search the lexicon for the form of that word that matches the verb in singularity, plurality and grammatical person. If the sentence were generated using only the lemmas, we would do the same since we already have the lemma of the word. Finally, when a sentence is generated using synsets, as mentioned in Section 4.5 of Chapter 4, each of the words comprising the synsets are extracted. These words are usually in lemma form. Subsequently, the same process is applied here.

Once each of the words of the sentence are inflected, the module follows the process described in the previous chapter (see Section 4.5 of Chapter 4).

Spanish Inflection

Regarding the Spanish inflection, as in the case of English inflection, the Spanish lexicon within the Freeling tool (Padró & Stanilovsky, 2012) is used. The process to inflect the words is the same as the one performed for English, with the exception of the verb inflection. In this regard, if the desired verb tense is not available in the lexicon, we would predict the inflection of this concrete verb tense using supervised learning.

Feature	**Description**
(1) *ending*	ending of the verb that can be “-ar", “-er" and “-ir", used to classify the verbs in 1st, 2nd, and 3rd conjugation respectively.
(2) *ending stem*	the closest consonant or group of letters to the ending, being part of the same syllable of the ending.
(3) *penSyl*	the previous syllable of the ending, consisting of the whole syllable or the dominant vowel.
(4) *person*	grammatical distinction between references to participants in an event, which can be 1st (the speaker), 2nd (the addressee) and 3rd (others) person.
(5) *number*	grammatical category that expresses count distinctions, which can be singular (one) or plural (more than one).
(6) *tense*	category that expresses time reference, in Spanish there are 17 different verb tenses.
(7) *mood*	grammatical features of the verbs used for denoting modality (statement of facts, of desire, of commands, etc.), in Spanish there are three different moods.
(8) *suff1*	one of the possible inflections for the ending.
(9) *suff2*	one of the possible inflections for the ending.
(10) *stemC1*	one of the possible inflections for the stem.
(11) *stemC2*	one of the possible inflections for the stem.
(12) *stemC3*	one of the possible inflections for the stem.

Table 5.1: Detailed description of the features used in the supervised learning approach. A specific verb tense in Spanish can have more than one valid inflection, being necessary to predict each variant of the tense.

In order to predict a verb infection, an ensemble of models was trained using the dataset described in (Barros et al., 2017b). This dataset is composed of the following features: (1) *ending*, (2) *ending stem*, (3) *penSyl*, (4) *person*, (5) *number*, (6) *tense*, (7) *mood*, (8) *suff1*, (9) *suff2*, (10) *stemC1*, (11) *stemC2* and (12) *stemC3* (see Table 5.1).

In the construction of the Spanish configuration of the inflection module, we consider that a verb can be divided into three distinct parts as shown in Figure 5.1: (i) *ending* (i.e., the verbs are classified by their ending in Spanish); (ii) *ending stem* (i.e., the consonant which is more close to the ending) and (iii) *penSyl* (i.e., the previous syllable of the ending).

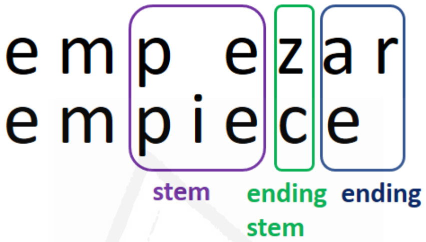


Figure 5.1: Division of the Spanish verb *to begin* and its inflection for the first singular person of the present tense and in the subjunctive mood.

In this dataset, the features *Suff1* and *suff2* represent the inflection predicted for the suffix of the verb; while the *stemC1*, *stemC2* and *stemC3* features refer to the inflection predicted for the stem of the verb. Therefore, we trained a different model for each of the features with a potential inflection value. Specifically, we used the Random Forest algorithm implementation in WEKA (Frank, Hall, & Witten, 2016) to train the models for the *stemC3* and *stemC2* features, and the Random Tree algorithm implementation provided also in WEKA for training the models for the *suff1*, *suff2* and *stemC1* features. These algorithms were the ones that delivered the better results after testing all the algorithms in WEKA.

Once the models are trained, we analyze the base form the verb, i.e. its lemma, to extract the necessary features (i.e. the suffix and the stem of the verb) for its inflection. Then, we predict the inflection of each of these features employing the trained models separately. Finally, we create the inflection of the verb by reconstructing the lemma replacing the features extracted with their corresponding inflections (see Figure 5.2).

The remainder of the inflection process is performed as described in the previous chapter (see Section 4.5 of Chapter 4).

5.1.3 Scenarios

For the purposes of this research work, some scenarios were proposed to verify the flexibility of HanaNLG. In this sense, we will consider two distinct scenarios: i) NLG for assistive technologies and ii) NLG for opinionated sentences. These

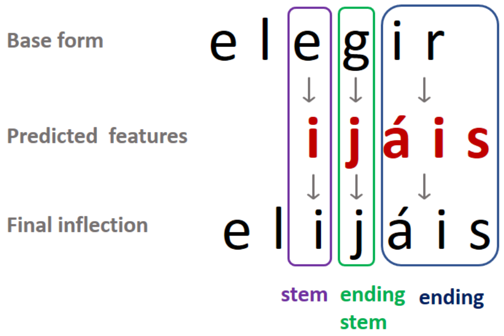


Figure 5.2: Reconstruction of the verb *elegir* (to choose) with the features predicted by the models.

scenarios were selected due to their diversity in purpose and theme, each of them being related to one of the aforementioned seed features described.

In addition to these scenarios, there is a third scenario which is related to the last type of seed feature (i.e., keywords or important words) defined in Section 5.1.1. In this sense, this scenario is the generation of topic-focused texts and is assessed within the context of the headline generation application. However, since the assessment of this application is considered to be an extrinsic evaluation, it will be discussed in Chapter 6.

NLG for Assistive Technologies

Our first scenario is focused on the generation of sentences that will be useful in the domain of assistive technologies. In this respect the main objective of this scenario is the creation of sentences for the dyslalia disorder. Dyslalia is a disorder that affects to the phoneme articulation. This implies the inability of correctly pronounce certain phonemes or a group of phonemes. This disorder usually affects the child population, with an approximate a 5%-10% incidence (Conde-Guzón et al., 2014).

Based on this scenario, the objective of the experimentation is to generate sentences containing words with a concrete phoneme. These types of sentences have demonstrated their usefulness in dyslalia speech therapies (Rvachew et al., 1999). Subsequently, the type of seed feature employed for this scenario is phoneme.

NLG for Opinionated Sentences

Recently, the use of Web pages where people express their opinions (such as Amazon[3], RottenTomataoes[4] or TripAdvisor[5]) has increased. These opinions are usually stated in the form of visual numeric ratings (such as stars) or textual reviews. In the case of ratings, the information provided is limited and sometimes difficult to interpret. The generation of text that may explain these ratings would be useful for the users of these pages.

Therefore, based on this, our second scenario is focused on the generation of sentences with positive or negative polarity. In this regard, the experimentation is centered on the movie review context. Thus, the main objective is to generate positive or negative sentences related to a movie, with polarity being used as the seed feature for HanaNLG.

5.2 Evaluation Research Questions

As stated in the introduction, some issues were taken into account in the construction of HanaNLG. These issues will be further analyzed and discussed in the following research questions. Therefore, we will perform an incremental evaluation where the answers to one question will justify the decisions made in subsequent questions.

- **Does the use of seed features enable the generation of language which can fulfill a specific purpose or target?** With this question we want to check to what extent the use of seed features within a statistical generation process provides a means for generating language capable of addressing a specific purpose or target.

- **Which type of language model is more appropriate for generating language in the architecture of HanaNLG?** The goal of this research question is to analyze two types of LM (i.e., n-grams and FLM) in order to determine which one is more appropriate for their use in the Sentence generation and Sentence ranking modules of HanaNLG.

- **In the event that FLM performs better than n-grams, what kind of factors provide more flexibility, in terms of content, when generating language?** The main objective of this research question is to analyze diverse features for building the FLM used in the generation, and determine which ones provide more flexibility with respect to the content generated.

- **Does the use of seed features in conjunction with factored language models allow the generation of language for different domains and languages?**

[3] https://www.amazon.com/
[4] https://www.rottentomatoes.com/
[5] https://www.tripadvisor.com/

The results obtained from this analysis may provide insights about the appropriateness of the methodologies employed for generating language in different contexts and languages.

- **To what extent does the integration of an inflection module improve the naturalness and expressivity of the generated language?** With this research question we wanted to test the performance of the inflection module of HanaNLG.
- **Does the combination of semantic and syntactic information affect the quality of the text generated?** This last research question comprises the evaluation of HanaNLG as a whole approach with the architecture of modules detailed in Chapter 4.

These research questions allow us to extensively assess every individual and global aspect of HanaNLG. The following sections describe and answer each of the research questions proposed.

5.3 Analysis of the Use of Seed Features within a Statistical Natural Language Approach for Fulfilling a Specific Purpose or Target

Seed features provide flexibility in terms of the content in the text generated, and therefore, are one of the main aspects underpinning our approach. Thus, in a preliminary evaluation, we evaluated the suitability of the seed feature for generating a text that fulfils the requirements of a specific purpose or target.

In order to conduct this analysis, we built a basic statistical approach for the surface realization stage whose main objective was to generate sentences with the highest number of words related to a specific seed feature given as input. In this sense, we used n-grams in conjunction with input seed features for generating the sentences. We will refer to this approach as HanaNLG$_{N-gram}$.

Each sentence was generated as follows. First, an n-gram LM was trained over a corpus given as input. Second, a bag of words is obtained containing words related to a specific seed feature. Finally, the sentences are generated following an iterative process, which is repeated until a desired length (entered as input parameter) or the special token “end of sentence” (</s>) is reached. So, starting from the special token “start of sentence” (<s>), in each iteration, we chose the word related to the input seed feature and with the highest probability of appearing after the word predicted in the previous iteration.

5.3.1 Scenario, Corpus and Resources

We place this analysis in the context of the assistive technologies scenario described in Section 5.1.3. In particular, we focused the experimentation on the generation of sentences with a particular phoneme for the Spanish language.

With this aim, we used a collection of Hans Christian Andersen stories[6] in Spanish. This collection is composed of 158 children's stories, containing a total of 21,085 sentences. Table 5.2 summarizes some statistics related to this collection.

Language	**Corpus**	**Files**	**Sentences**	**Avg Sents. per file**	**Words**	**Avg Words per file**
ES	H.C. Andersen	158	21310	134	339443	2148

Table 5.2: Statistics of the Spanish Hans Christian Andersen children stories corpus.

We used the 25% of this collection for obtaining the words of the bag of words (the ones related to the seed feature) and the remainder of the collection (i.e., 75% of the collection) for training the n-gram LM, which in our case involved training bigrams and trigrams. This is due to the fact that bigrams and trigrams are the most commonly used when handling n-grams (Rosenfeld, 2000). The tool SRILM (Stolcke, 2002) was used for building and training these LM.

5.3.2 Experimentation

Since our main objective is to analyze the appropriateness of the seed features for fulfilling a specific purpose or target, we generated several sentences for each of the Spanish phonemes (i.e., 27 phonemes). For building the LM, the punctuation marks were removed for simplifying the generation process.

5.3.3 Results and Discussion

HanaNLG$_{N-gram}$ generated a total of 208 sentences. Among them, 119 sentences contain the special token "end of sentence" (</s>) as their final element. The sentences containing this special token are important since they can be comparable to a complete sentence. This is because this token indicates when a sentence ends, and it usually appears within the language model trained after a full stop. Therefore, the evaluation and analysis of the results obtained will only be focused on these sentences. Moreover, since all the sentences have been generated using trigrams and bigrams, it may occur that the same sentence was generated by both LM. So, from the 119 produced sentences, 95 of them are completely different.

In order to assess these sentences, some statistics were calculated with respect to the input corpus. In addition, we manually evaluated the meaningfulness of the sentences, considering that a sentence may become meaningful with the inclusion of some punctuation marks, since they were removed during the training of the LM. Table 5.2 summarizes the statistics obtained for the sentences generated by HanaNLG$_{N-gram}$. These statistics were computed on the basis of

[6] http://www.ciudadseva.com/textos/cuentos/euro/andersen/hca.htm

the total number of different sentences ending with the token “end of sentence” (</s>).

Sentences	**Local percentage (based on 95 sent.)**	**Global percentage (based on 208 sent.)**
Generated sent. from bigram LM with (</s>)	46.32%	21.15%
Generated sent. from trigram LM with (</s>)	78.95%	36.06%
Newly generated not included in the corpus	73.68%	33.65%
Meaningful total sentences	56.84%	25.96%
Meaningful sentences included in the corpus	25.26%	11.54%
Newly meaningful generated sent. not included in the corpus	31.58%	14.42%
Newly meaningful generated sent. from bigram LM	9.47%	4.33%
Newly meaningful generated sent. from trigram LM	22.11%	10.10%

Table 5.2: Statistics of the sentences generated by HanaNLG$_{N-gram}$ which ended with (</s>).

As indicated by the Table, the results from the meaningful sentences are encouraging. Almost half of the sentences that ended with the special token are meaningful. Moreover, the meaningful sentences represent about 30% of the 95 different sentences concerning the ones that do not explicitly appear in the input corpus. Furthermore, we checked that every single generated sentence contained at least one word related to input seed feature. These results are quite positive and show that the use of seed features within a simple generation process can lead to meaningful sentences that fulfill a concrete purpose or target. Example 14 shows some of the newly generated sentences, with their English translation in brackets.

(14) **Phoneme:** /a/ **Sent:** *<s>allí se quedó con la doncella había llegado el invierno </s>*
(stayed there with the maid the winter had come)
Phoneme: /f/ **Sent:** *<s>finalmente llegaron a la superficie del agua </s>*
(they finally reached the surface of the water)
Phoneme: /e/ **Sent:** *<s>pues bien hecho está </s>*
(well done)
Phoneme: /n/ **Sent:** *<s>dónde está el cielo </s>*
(where is the sky)

5.4 Analysis of the Appropriateness of Language Models for Generating Language

Given that the first research question, detailed in the previous section, obtained promising results with the use of n-grams as LM, the objective of the second research question is to analyze other LM for the generation process.

Thus, we will analyze the performance of FLM in contrast to the use of n-grams in the generation process. Therefore, we will compare the results obtained from this analysis to the ones from HanaNLG$_{N-gram}$.

The procedure to construct a sentence using FLM, in this experiment, was the same as that used in the case of HanaNLG$_{N-gram}$ (see Section 5.3). However, some aspects of the selection of words differed. In this respect, the words predicted in each iteration were selected based on the FLM's factors. In addition to this, the stop condition of the iterative process was also different. In this sense, the selection of new words concluded when a full stop or the maximum length of the sentence was reached. In addition to this, in this experimentation, in the case of FLM, a ranking was not performed at the end of the generation process to select only one sentence. We will refer to this approach using FLM (regardless of the factors used during its training) as HanaNLG$_{FLM}$.

5.4.1 Scenario, Corpus and Resources

For the purpose of this experimentation, we placed the analysis of the generated sentences in the same context as that used for the first research question. Therefore, we selected the assistive technologies scenario for testing the FLM. Moreover, in order to compare more equitably both types of LM, we decided to test them in two distinct languages: English and Spanish.

Concerning the corpus used as input for both approaches, we used the same collection of Hans Christian Andersen stories for the generation of sentences in Spanish and, in the case of English, we gathered the same collection of stories[7] in this language[8]. In addition, we calculated the average sentence length in the corpus, which was, in this case, 16. Table 5.3 summarizes the statistics of both collections.

Language	Corpus	Files	Sentences	Avg Sents. per file	Words	Avg Words per file
EN	H.C. Andersen	140	6246	44	11278	80
ES	H.C. Andersen	158	21310	134	339443	2148

Table 5.3: Statistics of the English and Spanish Hans Christian Andersen children stories corpora.

[7] Some files of this corpus may contain more than one tale per file.

[8] http://hca.gilead.org.il/

Regarding the tools used in this experimentation, the software SRILM(Stolcke, 2002) was used to build and train both LM, n-grams and FLM. The tool Freeling (Padró & Stanilovsky, 2012) was used to analyze the input corpus for obtaining some of the factors needed in the FLMs.

5.4.2 Experimentation

The experimentation in this research was focused on the analysis of whether the use of FLM could lead to the generation of better sentences in contrast to the ones generated by HanaNLG$_{N-gram}$. In addition to this, we also wanted to test the performance of both LM in different languages. Therefore, as previously mentioned, English and Spanish were selected. So, we generated several sentences using the phonemes from both languages, being 44 the total number of phonemes in English and 27 the number of phonemes in Spanish.

Regarding the factors used for training the FLM, we used three distinct factors: (i) the word itself, (ii) its lemma ,and (iii) its POS tag. We tested two FLM with the following combination of factors: (i) Word+POS tag (*WP*) and (ii) Lemma+POS tag (*LP*). These factors were selected based on the type of information they provide. When using the combination *WP*, the POS tag adds information about the syntactic structure of the corpus on which the model was built. In the case of the *LP* configuration, besides this syntactic information, lemmas add a level of abstraction since they can be inflected to form different words.

For the purpose of this experimentation, and, taking into account that in the evaluation conducted in Section 5.3, trigrams obtained better results than bigrams, and therefore, we decided to use the former for building both n-grams and FLM.

5.4.3 Results and Discussion

Since, in this experiment we are only generating sentences without applying any type of ranking, we will only evaluate the produced sentences that ended with a full stop. In this regard, a total of 140 and and 95 sentences using HanaNLG$_{N-gram}$, and a total of 33 and 64 sentences employing HanaNLG$_{FLM}$ were generated for English and Spanish respectively.

To evaluate these sentences, a manual evaluation with two assessors was performed. These assessors were volunteer students from different bachelors with a proficiency level of English. In this manual evaluation the assessors were asked to discern if the sentences were meaningful or not. Table 5.4 summarizes the results obtained from this manual evaluation. In addition, the table also shows the percentage of sentences that do not explicitly appear in the training corpus as well as the percentage of meaningful sentences that are in the corpus. As seen in the table, the results obtained by any of the two configurations of HanaNLG$_{FLM}$ surpasses almost every result from HanaNLG$_{N-gram}$, verifying its appropriateness.

Language Model		**Total generated sentences**	**Meaningful generated sentences**	**Newly meaningful sent. (not in corpus)**	**Meaningful sentences in corpus**
EN	HanaNLG$_{N-gram}$	140	51.43%	34.29%	17.14%
	HanaNLG$_{FLM}$ *WP*	21	33.33%	28.57%	4.76%
	HanaNLG$_{FLM}$ *LP*	33	75.75%	72.72%	3.03%
ES	HanaNLG$_{N-gram}$	95	56.84%	31.58%	25.26%
	HanaNLG$_{FLM}$ *WP*	67	77.61%	53.73%	23.88%
	HanaNLG$_{FLM}$ *LP*	64	79.69%	54.69%	25%

Table 5.4: Results and comparison of the FLM employed in HanaNLG$_{FLM}$ with respect to HanaNLG$_{N-gram}$ for the assistive technologies scenario.

In the case of Spanish, the results from HanaNLG$_{FLM}$ have improved with regard to the HanaNLG$_{N-gram}$ ones. However, the English ones do not seem to improve much. The reason behind this is that most of the sentences generated in English are really long and may lack meaning due to grammatical errors. In addition, the number of sentences generated in English has decreased in contrast to the ones generated in Spanish. This is because, when the first word after the token start of sentence (<s>) is selected, this word is determined based on the token start of the sentence and the most probable POS tag appearing after this token. In the case of English, the most probable POS tag is a pronoun, therefore, very few sentences beginning with a pronoun and ending with a full stop are generated. Example 15 shows some sentences generated in English and Spanish using the *LP* configuration of HanaNLG$_{FLM}$.

(15) **Spanish**
Phoneme: /o/
Sentences:
Conocer el bosque de abeto.
(Getting to know the fir forest.)
Contestar el viejo Ãąrbol.
(Answer the old tree.)
Dormir dulcemente.
(Sleep sweetly.)
Observar el padre tocar uno redoble en todo direcciÃşn.
(Watch the father play a roll in all directions.)
Resonar por todo el mundo.
(Resonate all over the world.)

English
Phoneme: /m/
Sentences:
Many thank for the moon become calm and still high mountain seem to come home.
My mother be asleep.

5.5 Analysis of the Flexibility Provided by Different Types of Factors, in Terms of Content

The main objective of this experimentation is to test several factors for the FLM used during the generation and determine which one can provide more flexibility in terms of content.

For producing the sentences in this experiment, the approach HanaNLG$_{FLM}$ (which was defined in the previous section) was used with the addition of a ranking stage. Therefore, this approach is based on over-generation and ranking techniques, where a ranking is performed after the generation of sentences, in order to select one. This ranking is the one used in the Sentence ranking module of HanaNLG (see Section 4.4 of Chapter 4). We will refer to this approach as HanaNLG$_{FLMR}$.

5.5.1 Scenario, Corpus and Resources

Since we only wanted to test the performance of the different factors, we did not choose any of the scenarios defined in Section 5.1.3. For the same reason, in this experimentation we did not rely on the use of seed features for the generation of language.

A collection of 779 English children stories was used as the corpus, so we will focus on the generation of sentences in the context of children fairy tales in English. This collection includes 264 English children stories automatically gathered from *Bedtime Stories*[9], the Lobos and Matos corpus (Lobo & De Matos, 2010) and the Hans Christian Andersen English collection previously mentioned. Table 5.5 summarizes some statistics from these corpora.

Language	Corpus	Files	Sentences	Avg Sents. per file	Words	Avg Words per file
EN	Lobos and Matos	376	15684	41	620264	1649
	Bedtime Stories	264	5029	19	114241	432
	H.C. Andersen	140	6246	44	11278	80

Table 5.5: Statistics of the corpus included in the collection of English children stories used as the corpus.

Concerning the tools used in this experimentation, the tools employed in the previous research questions (see Section 5.4) were used. In this sense, the language analyzer tool Freeling (Padró & Stanilovsky, 2012) was used to obtain linguistic information for constructing the model. Concretely, information about the word itself, its lemma, its POS tag and its synsets was gathered. In addition, for handling the synsets, we used the software JWI (Finlayson, 2014) in conjunction with WordNet (Fellbaum, 1998).

[9] https://freestoriesforkids.com/

5.5.2 Experimentation

The results obtained from the evaluation conducted in Section 5.4 yielded good results compared to the ones from HanaNLG$_{N-gram}$. However, only two types of factors were tested. Therefore, the main objective of this experimentation was to test what type of factors can provide more content flexibility in the generation of a sentence.

The factors used in this experiment are: (i) words; (ii) lemmas; (iii) POS tags; and (iv) synsets. Using these factors, we built and trained three different FLM with the following combination of factors: (i) Word+POS tag (*WP*); (ii) Lemma+POS tag (*LP*) and (iii) Synset+POS tag (*SP*). These factors were chosen due to several reasons. As previously mentioned, the use of the POS tags in the definition of the models provides information about the structure of the corpus used in the training of the model. Both lemmas and synsets raise the degree of abstraction in terms of content. In the case of the lemmas, they can be inflected to obtain words related to them (e.g., when the lemma of the verb "go" is inflected, the words "went", "going" or "gone" can be obtained). Concerning the synsets, they can be expanded to each of the words comprising them. Therefore, the variability in content, in contrast to the lemmas is higher when using synsets.

For each of the FLM trained, 20 sentences were generated, obtaining a total of 60 automatically produced sentences.

Concerning the ranking carried out, as mentioned in Section 4.4 of Chapter 4, we employed the following linear combination of FLM: $P(w_i) = \lambda_1 P(f_i|f_{i-2}, f_{i-1}) + \lambda_2 P(f_i|p_{i-2}, p_{i-1}) + \lambda_3 P(p_i|f_{i-2}, f_{i-1})$, where f can be either a word, a lemma or a synset; p refers to a POS tag, and λ_i are set to $\lambda_1 = 0.25$, $\lambda_2 = 0.25$ and $\lambda_3 = 0.5$. The values used for the λ were the ones that obtained the best results after testing several combinations of values.

5.5.3 Results and Discussion

In order to assess the sentences generated in this experiment, a user-based collaborative evaluation with a total of 12 assessors was carried out following the guidelines established in (Gkatzia & Mahamood, 2015). These assessors were volunteer students from different bachelor degrees with a proficiency level of English. We measured the agreement between the assessors with the kappa statistics implementation in (Randolph, 2008). An overall agreement of 0.58 (i.e., moderate agreement) was obtained, where each sentence was evaluated by at least 2 assessors. The reason that the agreement is moderate is due to the subjectivity of this type of evaluation.

This evaluation was carried out with the use of questionnaires, using a 5-pt Likert scale, which is a type of assessment frequently used in this arena, as mentioned in Section 3.2.4 of Chapter 3. In these questionnaires, the assessors were asked to measure the **coherence**, **usefulness**, **grammatical errors** and **structure** of the generated sentences. Specifically, in this experimentation, when we talk about the *coherence*, this term refers to what extent a sentence is meaningful as it is. For *coherence*, meaningless sentences would score 1 and the ones with a full coherent meaning a 5. In the case of *usefulness*, this term refers to the level of usefulness of the generated sentences in the context of them helping a writer create a children's story. As in the previous case, the value 1 is assigned to a sentence that would not be useful and 5 otherwise. Concerning the *grammatical errors*, this refers to the quantity of errors that the sentence contains. These errors can be related to the grammar, concordance or omission of some type of word, such as prepositions. Therefore, in this category, a sentence with many errors would be rated with the value 1 and with the value 5 in the case that the sentence has no error. Finally, the *structure*, refers to how well the structure of a sentence is in terms of syntax. A sentence with a lack of structure would be given a score of 1 and a well-formed sentence would be given a 5.

Factors	**Coherence**	**Usefulness**	**Grammar Errors**	**Structure**
HanaNLG$_{FLMR}$ *WP*	2.68	2.80	2.83	3.22
HanaNLG$_{FLMR}$ *LP*	3.08	3.31	3.00	3.53
HanaNLG$_{FLMR}$ *SP*	2.85	3.02	3.08	3.53

Table 5.6: Results of the means of the 5-pt Likert scale with respect to the coherence, usefulness of the sentence, grammatical errors and structure, of the sentences generated with FLMs.

Table 5.6 summarizes the mean results from this evaluation. The table shows that the results obtained when employing both configurations, *LP* and *SP*, outperforms the ones from the *WP* configuration for each of the criteria assessed. These results show that the use of more abstractive factors can lead to stronger and more expressive models. This demonstrates that these types of factors increase the flexibility of the language generated in terms of content. The average number of sentences derived from the ratings for each of the criteria assessed — *coherence*, *usefulness*, *grammatical errors* and *structure* — is shown in Figure 5.3.

As can be observed in the figure, the results of the *LP* and *SP* configurations of HanaNLG$_{FLMR}$ are very similar in terms of their average ratings. However, the latter can provide more flexibility and expressive richness to the generated language. This is because the synsets contain more semantic information and can be expanded into different synonyms depending on the context. Example 16 illustrates each of the configurations explored.

(16) ***HanaNLG$_{FLMR}$* WP:** *Some time passed the night.*
***HanaNLG$_{FLMR}$* LP:** *A little boy was the king.*
***HanaNLG$_{FLMR}$* SP:** *The whole town knew the good thing.*

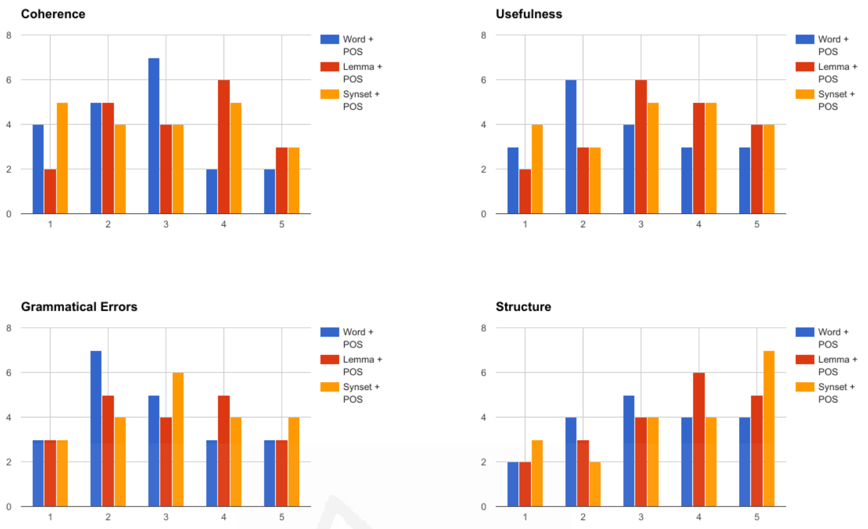


Figure 5.3: Number of sentences scored for each rating of the 5-pt Likert scale regarding the *coherence*, the *usefulness*, the *grammatical errors* and the *structure*. The minimum values for the *coherence* indicate a lack of meaning of the sentences whereas the maximum values indicate a correct full meaning for a sentence. For the *usefulness* factor, the ratings were measured in the context of the usability of the sentences in storytelling, being the minimum value for sentences not usable in the context and the maximum value the opposite. For the *grammatical errors* ratings, the minimum values represents a high number of errors within the sentence while the maximum values indicates a lack of errors in the sentence. Finally, the lower values for the *structure* indicates a lack of structure in the sentence, and the higher ones, indicates a well-formed structure. Axis X represents the rating received for a sentence and axis Y represents the number of sentences that received that rating.

5.6 Analysis of the Performance of Seed Features in Conjunction with Factored Language Models for Different Domains and Languages

The use of seed features and FLM individually have yielded good results in previous sections. In addition to this, their combination has also obtained promising results in the case of a single domain. Therefore, the main objective of this experimentation is to test the performance of this combination in different domains and also for different languages.

For conducting this analysis we will employ the approach HanaNLG$_{FLMR}$, which addresses the surface realization stage using FLM and over-generation and ranking techniques. Since there is no change in the technique to automatically produce language, in this experimentation, we will also refer to the approach as HanaNLG$_{FLMR}$.

5.6.1 Scenario, Corpus and Resources

We focused the experimentation on the generation of sentences for the two scenarios defined in Section 5.1.3 — NLG for assistive technologies and NLG for opinionated sentences — in two different languages: English and Spanish. In the case of our first scenario, the collection of fairy tales of Hans Christian Andersen in Spanish and English was used as corpora. Concerning our second scenario, we selected two well-known movie review corpus. In this regard, the Sentiment Polarity Dataset (Pang & Lee, 2004) was used as corpus for English. In the case of Spanish, the Spanish Movie Reviews corpus[10] were employed as corpus. Table 5.7 outlines some of the statistics of the corpora from both scenarios.

Language	Corpus	Files	Sentences	Avg Sents. per file	Words	Avg Words per file
EN	H.C. Andersen	140	6246	44	11278	80
	Sentiment Polarity Dataset	2000	64720	32	1492663	746
ES	H.C. Andersen	158	21310	134	339443	2148
	Spanish Movie Reviews	3878	67622	17	1921855	495

Table 5.7: Statistics of the corpora employed in the experimentation for English and Spanish.

Regarding the tools used in this experimentation, some of the tools previously mentioned in Section 4.7 of Chapter 4 were employed. In this sense, Freeling (Padró & Stanilovsky, 2012) was used to obtain information about the factors used in the construction of the FLM. These FLM were built and trained using the software SRILM (Stolcke, 2002). Finally, we used the sentiment analysis classifier described in Fernández et al. (2013) to verify that the sentences generated correspond to the intended polarity.

5.6.2 Experimentation

The experimentation in this research was focused on the analysis of the combination of seed features with FLM to generate sentences in different scenarios and languages. With this aim, as previously mentioned, we selected two distinguishable scenarios — NLG for assistive technologies and NLG for opinionated sentences — and the following languages: English and Spanish.

[10] http://www.lsi.us.es/ fermin/corpusCine.zip

For constructing the sentences we decided to employ the configuration Lemma+POS tag (*LP*) and trigrams for building the FLM. This configuration was selected due to the good results obtained in the previous experiments.

Since the experimentation is focused in the two proposed scenarios, several sentences were generated according to the characteristics of each of them. In this sense, a total of 44 and 27 sentences (i.e. this is the total number of phonemes in English and Spanish respectively) were generated for our first scenario and for the second scenario, a total of 2 sentences (one negative sentence and one positive sentence) for each language.

Regarding the ranking, we employed the ranking module within HanaNLG (see Section 4.4 of Chapter 4) with the following linear combination of FLM: $P(w_i) = \lambda_1 P(f_i|f_{i-2}, f_{i-1}) + \lambda_2 P(f_i|p_{i-2}, p_{i-1}) + \lambda_3 P(p_i|f_{i-2}, f_{i-1})$, where f is a lemma , p refers to a POS tag, and λ_i are set $\lambda_1 = 0.25$, $\lambda_2 = 0.25$ and $\lambda_3 = 0.5$. These values correspond to the ones used in the previous section. They were determined empirically by testing different values and comparing the results obtained.

5.6.3 Results and Discussion

In order to assess the sentences generated in this experimentation, a manual evaluation with three different assessors was performed to verify the meaningfulness of these sentences. The assessors were asked to discern if a sentence is meaningful by taking into account several issues: i) if the sentence is meaningful by itself; ii) if the sentence becomes meaningful by adding some punctuation marks; and, (iii) if the sentence acquires meaning by inserting a preposition that usually follows the main verb.

We measured the agreement between the assessors using the kappa statistic implemented in (Randolph, 2008). A good agreement was obtained in both scenarios: an overall agreement of 0.83 and 0.78 for the first scenario in English and Spanish respectively; and an overall agreement of 1 in the case of the second scenario in both languages. Each sentence of this experiment was evaluated by at least 2 assessors.

Table 5.8 summarizes the results obtained from the manual evaluation. As seen in the table, the results obtained using HanaNLG$_{FLMR}$ to generate text are promising. However, since the principal factor employed within the FLM trained is lemma, the sentences generated do not contain inflected words. In addition to this, as a result of the evaluation carried out, the resulting sentences may not be strictly correct and may contain some errors.

In addition to this, the generated sentences were evaluated to verify if they met the characteristics of each scenario. This was done by calculating the percentage of words containing a specific phoneme with respect to the total length of the sentence in the case of our first scenario. Concerning the second scenario, we used the sentiment analysis classifier mentioned before. In both scenarios, the sentences met the all the characteristics of the scenarios proposed.

Scenario		**Meaningful generated sentences**	**Newly meaningful sent. (not in corpus)**	**Meaningful sent. with seed features**
EN	NLG for assistive technologies	95%	70%	82.5%
	NLG for opinionated sentences	100%	50%	50%
ES	NLG for assistive technologies	88.89%	40.74%	88.89%
	NLG for opinionated sentences	100%	100%	100%

Table 5.8: Comparative table of the sentences generated by HanaNLG$_{FLMR}$ for the two proposed scenarios.

Some examples of the sentences generated by HanaNLG$_{FLMR}$ in English and Spanish for both scenarios are shown in Example 17.

(17) **Opinionated NLG**
Polarity: Positive **Sent:** *The good work in this respect.*
Polarity: Negative **Sent:** *The acting be horrible .*
Polarity: Negative **Sent:** *Su falta de imaginación. Trans: Their lack of imagination.*

NLG for assistive technologies
Phoneme: /m/ **Sent:** *My mother be asleep.*
Phoneme: /b/ **Sent:** *I be bear in the book of fairy tale.*
Phoneme: /k/ **Sent:** *Cantar el canción popular. Trans: Sing the popular song.*

5.7 Analysis of the Integration of an Inflection Module for Improving the Naturalness and Expressivity of the Generated Language

Morphological inflection is key for making the language as natural as possible. In this sense, the naturalness and expressivity of the language generated in a NLG system can be improved by enriching the language through its morphology. Therefore, the main objective of this experimentation is to test the performance of the inflection module within HanaNLG.

In order to achieve this objective, we will use the proposed inflection module, in its Spanish configuration (see Section 5.1.2), with the HanaNLG$_{FLMR}$ approach used in previous sections. The selection of this language instead of English is not trivial. This is because Spanish is a morphologically-rich language. In addition to this, we also wanted to test the generation of related sentences. Therefore, a simple grammar, based on the structure of subject-verb-object, is used to guarantee that some elements of the sentence appear. Figure 5.4 shows this simple grammar. This type of grammar would help us, as a first step, in the construction of related sentences in terms of subject and object of the sentence. We will refer to this approach as HanaNLG$_{INF}$.

$$\begin{gathered} S \rightarrow NP\ VP \\ NP \rightarrow D\ N \\ VP \rightarrow V\ NP \end{gathered}$$

Figure 5.4: Basic clause structure grammar.

5.7.1 Scenario, Corpus and Resources

The experimentation was focused on the assistive technologies scenario defined in Section 5.1.3. Therefore, we use the collection of Hans Christian Andersen tales in Spanish as the corpus in this experimentation. Table 5.9 summarizes some of the statistics of this corpus.

Language	**Corpus**	**Files**	**Sentences**	**Avg Sents. per file**	**Words**	**Avg Words per file**
ES	H.C. Andersen	158	21310	134	339443	2148

Table 5.9: Statistics of the Spanish Hans Christian Andersen children stories corpus.

With respect to the tools used in this experimentation, we used Freeling (Padró & Stanilovsky, 2012) to obtain information related to the words of the corpus. For building and training the FLM used for the generation and ranking of the sentences, the software SRILM (Stolcke, 2002) was employed.

5.7.2 Experimentation

Since the main objective of this research was to test the performance of the inflection module of HanaNLG, we carried out two distinct experiments. First, we tested the inflection module alone by comparing it against some state-of-the-art systems to assess accuracy for the task of morphological inflection. Second, we tested the performance of the inflection module in conjunction with HanaNLG$_{FLM}$.

In the case of the first experiment, we compared our inflection module with the following competitive baselines: Durret13 (Durrett & DeNero, 2013) and Ahlberg14 (Ahlberg et al., 2014). This was done by measuring the accuracy of the Spanish verb infection under the same conditions. In this sense, we used a test set, composed of 200 different base forms, for comparing our inflection module with the competitive systems. This test set was made available by Durrett and DeNero (2013).

Concerning our second experiment, we generated sets of three related sentences for each of the Spanish phonemes (i.e., 27 phonemes) using the Lemma+

POS tag configuration and trigrams in the construction of the FLM. The relation of these sentences is based on a topic that will appear in the set of sentences so that the object of a sentence is used as the subject of the following sentence. We tested two types of inflection configurations in this experimentation: random and fixed. In the former configuration, a random verb tense is assigned to each of the sentences comprising the set. In the case of the latter configuration, the verb tense for the whole set of sentences is fixed to a unique verb tense, such as present indicative.

5.7.3 Results and Discussion

This section details the results obtained from the two experiments described. First, the results from the comparison of the inflection module of HanaNLG with two competitive systems. And, second, the results obtained from the generation of related sentences with HanaNLG$_{INF}$.

Comparison with the State-of-the-Art

Table 5.10 summarizes the results obtained from the comparison of the inflection module of HanaNLG with the competitive systems, Durret13 and Ahlberg14. The results of this table show that our inflection module obtains a higher overall accuracy, in the case of Spanish, compared with the competitive state-of-the-art systems.

Approach	**Correctly predicted verb tables**	**Correctly predicted verb forms**
HanaNLG$_{INF}$	**99**%	**99.98**%
Durret13	97%	99.76%
Ahlberg14	96%	99.52%

Table 5.10: Accuracy of predicting inflection of verb tables and individual verb forms given only the base form, evaluated with an unseen test set of 200 verbs.

Evaluation of the Sets of Related Sentences

We carried out a user-collaborative evaluation with three assessors for assessing the sentences generated by HanaNLG. The objective of this evaluation is to discern if the naturalness and expressivity of the language improved with the use of an inflection module.

Each assessor was shown 27 sets of sentences using different inflections (i.e., sentences without any type of inflection, sentences with the fixed inflection configuration and sentences with the random inflection one). They had to overall score each set using a 5-pt Likert scale for several aspects: (i) *coherence*; (ii) *grammatical errors* and (iii) *post-editing*. Specifically, (i) *coherence* refers to the degree of meaningfulness of the generated sentence, being meaningless the sentences scored with a 1 and meaningful those with a 5. The aspect *grammatical errors* refers to the amount of errors that the sentence contains. Therefore, a sentence would score a 1 in this category when it has many errors and a 5 in the case that it has no errors. Finally, the term *post-editing* (ease of correction) indicates the amount of changes that a sentence would need in order to correct mistakes. In this case, a lower score would mean that the sentence needs many changes whereas a higher value implies the opposite.

Inflection Type	**Coherence**		**Grammar errors**		**Post-editing**	
	Mean	Mode	Mean	Mode	Mean	Mode
HanaNLG$_{FLMR}$ LP (Without)	2.65*	2	2.73*	3	2.75*	3
HanaNLG$_{INF}$ Fixed	**3.36***	3	**3.57***	3	**3.54***	4
HanaNLG$_{INF}$ Random	3.31*	5	3.51*	4	3.48*	4

Table 5.11: Results of the means and the modes of the 5-pt Likert scale with respect to the *coherence*, *grammatical errors* and *post-editing(ease of correction)*, of the inflected generated sentences. * denotes significance with $p < 0.01$.

Table 5.11 shows the means obtained from the manual evaluation. As expected, both of the inflection configurations achieve better results for each of the assessment aspects compared to not inflecting the sentence. These results indicate that the quality of the sentences improved with the inclusion of the inflection module. Figure 5.5 summarizes the average number of sentences derived from the evaluation for each of the inflection configurations explored. As seen in this figure, the sentences without inflection scored less than the ones with either of the inflection configurations. These results corroborate the ones obtained in Table 5.11, thus demonstrating the improvement of the quality of the sentence after the use of the inflection module.

Some examples of sets of related sentences are shown in Figure 5.5.

(18) ***Phoneme:*** */n/*
Without Inflection
Cuánto cosa tener nuestro pensamiento.
(How much thing to have our thinking.)
Cuánto pensamiento tener nuestro corazón.
(How much thought to have our heart.)
Cuánto corazón tener nuestro pensamiento.
(How much heart to have our thinking.)

Fixed Inflection
Cuánta cosa tiene nuestro pensamiento.
(How much thing our thinking has.)
Cuánto pensamiento tiene nuestro corazón.
(How much thought our heart has.)
Cuánto corazón tiene nuestro pensamiento.
(How much heart our thinking has.)

Random Inflection
Cuánta cosa tiene nuestro pensamiento.
(How much thing our thinking has.)
Cuánto pensamiento tuviere nuestro corazón.
(How much thought our heart had.)
Cuánto corazón tenga nuestro pensamiento.
(How much heart our thinking had.)

5.8 Analysis of the Quality of the Text Generated when Combining Semantic and Syntactic Information

In the previous sections, we have seen the experimentation conducted to verify the appropriateness of the underlying methodologies used in HanaNLG. Once this has been proven, we can now focus on the evaluation of the performance of HanaNLG as an entire approach, composed of all the modules described in Chapter 4. Therefore, we will analyze if the inclusion of semantic and syntactic information can increase the quality of the text generation. In order to perform this analysis, we assess the text generated in the two described scenarios. For generating this text, HanaNLG follows an over-generation and ranking perspective, as specified in Chapter 4, where a set of sentences with the same seed feature is previously generated and then ranked in order for the purpose of selecting one. In addition to this, the inclusion of semantic and syntactic resources in the generation process will provide the approach with greater flexibility in terms of vocabulary, and including new information not contained in the training corpus.

This experimentation will only be focused on the generation of language for English. This is due to the fact that during the implementation of our approach, as mentioned, some of the resources employed are language-dependent. In this regard, VerbNet is currently available for English, the language chosen to perform the experimentation. This is not the case of WordNet, which has been adapted to other languages. However, we cannot employ only the frames from WordNet to generate text since they do not provide enough information.

5.8.1 Scenario, Corpus and Resources

As mentioned before, we focused the experimentation on the scenarios described in Section 5.1.3. Therefore, different corpora were used depending on the scenario in which the text was generated. Specifically, in the case of the first proposed scenario — NLG for assistive technologies — we used as corpus a collection of 779 documents automatically gathered from *Bedtime Stories*[11], the Lobos and Matos corpus (Lobo & De Matos, 2010) and the H.C. Andersen corpus automatically gathered from *Hans Christian Andersen: Fairy Tales and Stories*[12]. Regarding

[11] https://freestoriesforkids.com/
[12] http://hca.gilead.org.il/

the second scenario, Sentiment Polarity Dataset (Pang & Lee, 2004) was used as corpus. Table 5.12 summarizes some statistics from these corpora.

Scenario	**Corpus**	**Files**	**Sentences**	**Avg Sents. per file**	**Words**	**Avg Words per file**
Assistive Technologies	Lobos and Matos	376	15684	41	620264	1649
	Bedtime Stories	264	5029	19	114241	432
	H.C. Andersen	140	6246	44	11278	80
Opinionated Sentences	Sentiment Polarity Dataset	2000	64720	32	1492663	746

Table 5.12: Statistics of the corpora used in the experimentation for the proposed scenarios.

With respect to the tools used in this experiment, they are the same detailed in Section 4.7 of Chapter 4.

5.8.2 Experimentation

For the purpose of this experiment, we took as an input to our approach: (i) the corpus previously described, (ii) a different seed feature for each of the scenarios (i.e. a phoneme for our first scenario and a polarity for the second one), (iii) 1 as the number of sentences (since we are only generating a sentence for each seed feature), (iv) "lemma" as the level of abstraction and (v) "past" as the verb tense. The selection of these inputs was deliberate. Our output will be compared with the one from HanaNLG$_{FLMR}$. The configuration for HanaNLG is as similar as possible to that used in the experimentation described in Section 5.6. The reason for comparison with HanaNLG$_{FLMR}$ is that there are no gold-standards in this research area to compare our output and, to the best of our knowledge, there are no other surface realization systems or approaches working with these specific scenarios. Therefore, we cannot compare our output to any system of the state-of-the-art.

Therefore, the main objective of this experimentation is to generate, in the case of the first scenario, a sentence for each of the English phonemes (i.e. English has a total of 44 as previously mentioned) and a sentence for each polarity (i.e. negative and positive) in the case of the second scenario. Regarding the ranking module, we employed the same setting in the experimentation and evaluation as the one in Section 5.6.

5.8.3 Results and Discussion

In order to assess the generated language, a manual collaborative user-based evaluation with three assessors was conducted. These assessors were volunteer students from different bachelor degrees with a proficiency level of English. In this case, the assessors were asked to evaluate if a sentence is meaningful with respect to its coherence and structure. The issues evaluated here differ from the ones in the previous experiment. In this evaluation we are not assuming that the generated text may contain any kind of error due to the lack of punctuation marks or prepositions. Henceforth, the produced text is assessed as it is. With the aim of measuring the agreement between the assessors, the Kappa statistic (Randolph, 2008) was used. In this sense, we obtained an overall agreement of 0.84, which indicates a strong agreement between the assessors. Each sentence of the experimentation was evaluated by at least 2 assessors.

System	**Scenario**	**Meaningful Sentences**	**Newly Generated Sentences**	**Meaningful sent. with seed features**
HanaNLG	NLG for assistive technologies	**97.73%**	**100%**	**100%**
	NLG for opinionated sentences	**100%**	**100%**	**100%**
HanaNLG$_{FLMR}$*	NLG for assistive technologies	95%	70%	82.5%
	NLG for opinionated sentences	100%	50%	50%

Table 5.13: Comparative results of the manual evaluation for the two scenarios proposed. *These results correspond to the ones in Table 5.8.

Table 5.13 summarizes the results obtained from the manual evaluation for HanaNLG in comparison with the ones from HanaNLG$_{FLMR}$. As seen in the table, HanaNLG achieves better results on the generation of text in both of the proposed scenarios. In this sense, these results improve each of the aspects evaluated, creating new meaningful language (i.e. which does not exist explicitly in the input corpus) and all of the sentences containing their respective input seed feature. These results are not surprising. In terms of content, the text produced by our approach contained more semantic information. The use of linguistic resources, such as VerbNet, allows us greater control over the generated content, each time choosing the suitable type of word (e.g. a person, an object or an animal) required by the verb. In addition to the content, the frames gathered from Verbnet and WordNet provide us with a structure for the sentence, removing the need of having a complex or computational costly grammar. Moreover, it is worth mentioning that the sentences produced by HanaNLG have been automatically inflected using the past simple tense, providing them with a greater naturalness.

In light of these results, it has been proven that a hybrid perspective for NLG can provide more flexibility and language quality, allowing the adaptation of the generation process to different purposes and scenarios. Some of the generated sentences by HanaNLG and HanaNLG$_{FLMR}$ are shown in Example 19.

(19) ***HanaNLG***

NLG for assistive technologies
Phoneme: /l/ **Sent:** *Hjalmar looked towards the sea.*
Phoneme: /g/ **Sent:** *The hero fought with Argus.*
Phoneme: /th/ **Sent:** *The youth thanked God for the sooth.*
Phoneme: /d/ **Sent:** *The guard drew him through the field.*

NLG for opinionated sentences
Polarity: Positive **Sent:** *The apostle deserved the praise.*
Polarity: Negative **Sent:** *The jellyfish killed the member.*

HanaNLG$_{FLMR}$

NLG for assistive technologies
Phoneme: /l/ **Sent:** *All be call the land of eternity and look like one flower in the world.*
Phoneme: /g/ **Sent:** *I think of her and say the neighbors wife.*
Phoneme: /th/ **Sent:** *Something like one little brother and the other they say that he think of them.*
Phoneme: /d/ **Sent:** *Her head and say the old lady.*

NLG for opinionated sentences
Polarity: Positive **Sent:** *The good work in this respect.*
Polarity: Negative **Sent:** *The acting be horrible.*

As can be seen in this comparative example, the language produced by HanaNLG is more natural and the content reflects more meaningfulness than the examples from HanaNLG$_{FLMR}$. The language generated by our approach is shorter, but this is due to the length of the frames. Conversely, although the language generated by HanaNLG$_{FLMR}$ is longer, its structure seems more confusing and causes the meaning of the sentence to be lost.

5.9 Conclusion

This chapter outlined the experimentation carried out to assess the language generated by HanaNLG along with its environment configuration. First, the different issues that can be configured in our approach were outlined. In this sense, several types of seed features can be used to generate text that can be adapted to the purpose or target of the language. In addition, the language module of HanaNLG can be adapted to handle the inflection of different languages. Moreover, different scenarios were proposed and described. Second, the experimentation carried out allowed us to assess each aspect involved in the development on HanaNLG. In this sense, several research questions were addressed, and some of these included: the use of seed features to generate language; the use of FLM; the flexibility of adaptation for the generation in different scenarios; and, the performance of HanaNLG as a complete hybrid surface realization approach.

From the experimentation and evaluation carried out throughout this chapter we have obtained good results in the generation of sentences that are easily adaptable to different scenarios and purposes, as well as to different languages. This was possible due to the use of seed features in conjunction with FLM which provides the necessary flexibility to generate this type of text. Furthermore, the use of morphological inflection techniques and the combination of syntactic and semantic resources have demonstrably improved the quality of the language generated.

To summarize, the main contributions of this chapter were to:

- **Assess the appropriateness of the methodologies proposed during the implementation of HanaNLG.** In particular, the use of the seed features for the flexibility of adaptation for different domains and languages. Moreover, the use of FLM and the analysis of the different factors employed in the sentence generation and ranking. Furthermore, the evaluation of the quality of the language produced after the inclusion of an inflection module.

- **Assess HanaNLG's abilities for generating text that can be easily adapted to different scenarios and purposes.** HanaNLG was evaluated in different scenarios and their results were compared to the ones from HanaNLG$_{FLMR}$.

Furthermore, the evaluation carried out in this chapter revealed the advantages and limitations of our approach. Concerning its advantages, it is worth mentioning that the use of seed features in the implementation of HanaNLG allows us to easily adapt the generated language to different domains and purposes. In addition to this, the use of seed features and the integration of semantic resources also provide flexibility in vocabulary selection when generating text. However, the sentences generated are short due to the structure of the gathered frames. The use of other types of resources that provide us with more descriptive information as well as a more complete structure of the sentence's components would help us to address this flaw. As previously mentioned, at this stage, some modules of our approach are language dependent. Therefore, other linguistic resources that would provide us with semantic and syntactic information for the target language are needed to adapt these modules to that target language.

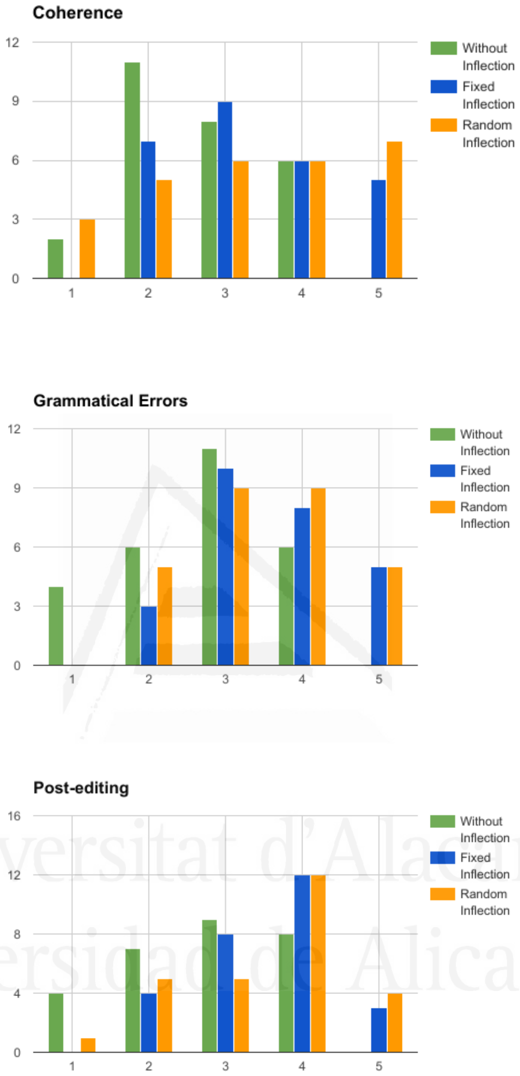


Figure 5.5: Number of sentences scored for each rating of the 5-pt Likert scale regarding the *coherence*, the *grammatical errors* and the *post-editing*. The minimum values for the *coherence* indicate a lack of meaning of the sentences whereas the maximum values indicate a correct full meaning for a sentence. For the *grammatical errors* ratings, the minimum values represent a high number of errors in the sentences and the higher values indicate a lack of errors in the sentences. Finally, the minimum *post-editing* values refers to a huge number of changes required to correct or improve the sentences while the maximum values indicates otherwise. Axis X represents the rating received for a sentence and axis Y represents the number of sentences that received that rating.

CHAPTER 6

HanaNLG Extrinsic Evaluation: Application in Automatic Summarization Tasks

The integration or the adaption of the techniques employed in NLG can be very useful for other NLP applications (e.g., automatic summarization) and also other AI areas. For instance, in the case of computer vision, NLG techniques could be employed in the generation of image captions which describe the content of an image (Bernardi et al., 2016). Or in case of computational creativity, these techniques can be used in the generation of jokes and puns (Gatt & Krahmer, 2018). When NLG techniques are used in other fields and applications, they can be evaluated extrinsically not only focusing on the generated text but also on how well the application's goal is met.

Therefore, the main objective of this chapter is to analyze the adaptability of HanaNLG to other applications of NLP. In particular, we tested HanaNLG under the text summarization field. Automatic summarization is the research field which aims to concisely reflect the most important information of a given document through the automatic creation of summaries (Nenkova & McKeown, 2011).

This task is usually approached in two different ways. On the one hand, extractive summarization analyzes the input text to determine the most important sentences in it and extracts them verbatim. On the other hand, abstractive summarization identifies the key information from a text and creates a new summary including new or different vocabulary, linguistic expressions or concepts not appearing in the original document. The latter, in comparison with the former, may lead to more natural human-like summaries. In this sense, NLG can provide useful techniques to facilitate the generation of abstractive summaries.

This chapter is structured in two sections, each of them addressing the analysis of HanaNLG in a different application within the text summarization field. First, Section 6.1 reports on the adaptation of HanaNLG for generating news article headlines. Then, Section 6.2 analyzes the generation of narrative cross-document timelines using an Enriched Timeline Extraction system in conjunction with HanaNLG. Finally, the main conclusions drawn are provided in Section 6.3.

6.1 HanaNLG for Headline Generation

Headline generation is one of the applications enclosed within automatic summarization and can be addressed from an extractive or an abstractive perspective. The objective of this application is to automatically produce a headline that describes the content of a news article. A headline is one of the most important parts of a news article, since, with just a glance, we can know what the news is about. Therefore, this headline represents the main idea of an article taking the form of a sentence or phrase.

The task of headline generation has been commonly addressed with extractive approaches. These approaches usually focus on the compression of the input document's sentences to obtain a headline. For instance, in (Dorr et al., 2003) one or two informative sentences are extracted and their length is reduced to form the headline. With respect to abstractive approaches, recently there has been an increase in headline generation approaches using this type of technique. Examples of abstractive summarization approaches for headline generation can be found in (Gatti et al., 2016) and in (Colmenares et al., 2019). In the former, the authors proposed an approach that combines an event-driven model with a multi-sentence compression algorithm to merge structural events for constructing the headlines. Regarding the latter, the authors present a sequence-prediction technique which models the problem of headline generation as a discrete optimization task in a feature-rich space. In this technique, the authors try to automatically learn how an editor discerns between good or bad headlines, which can be seen as a compression of their respective news article.

In this section, we outline the adaptation of HanaNLG to headline generation and we conduct an evaluation of the results obtained. As a starting point, we define the task to perform (Subsection 6.1.1), and further on, we report on the strategy followed to adapt HanaNLG for generating headlines (Subsection 6.1.2). Finally, we detail the experimentation carried out along with the evaluation of the headlines generated in comparison with the state-of-the-art.

6.1.1 Task Description

Headline generation, which has traditionally focused as a single-document summarization, is not an easy task. From an extractive perspective, the first sentence of the source article or the most important sentence within it can be considered to be the headline. This type of headline may not seem ideal since it may lack important facts from other sentences. Hence, a purely extractive approach is insufficient for creating a headline from a document (Banko et al., 2000). Alternatively, abstractive techniques are more appropriate to perform this task. This is due to the fact that these techniques analyze the input document, searching for key information in the text. Then, this information is paraphrased and combined into a single sentence, composing the final headline.

The task of abstractive sentence summarization was formalized in the Document Understanding Conferences (DUC), specifically at the DUC 2003 and DUC 2004 shared task (Over et al., 2007). In these tasks, the participants were asked to generate a very short summary (approximately 10 words or ≤ 75 bytes) from each of the documents within a news article corpus. Therefore, our scenario for the experimentation carried out for the generation of headlines is placed in the DUC's tasks. Based on these tasks, our main objective is to generate a headline of 10 or less words summarizing each of the documents of the corpora employed in these shared tasks.

6.1.2 Adapting HanaNLG for the Generation of Headlines

In order to be able to generate headlines with HanaNLG, the use of a certain seed feature is essential. In this regard, the seed feature "*keyword or important word*" (see Section 5.1.1 of Chapter 5) is employed for this case. This seed feature allows the generation of sentences related to a specific keyword or topic. These topics must be relevant, so it is necessary to establish a method to determine the relevant topics from which the sentence is going to be generated. Hence, this will allow the generation of headlines based on the essential information of the source documents.

When dealing with news articles, the essential information contained in them may be expressed with some specific type of terms or relevant elements. These elements may be verbs, nouns, adjectives or more complex structures. The detection of such relevant elements is done through the application of diverse heuristics. Specifically, several heuristics, that are based on techniques employed in automatic summarization, were tested, and, depending on the heuristic used a threshold may be needed to determine if a term is relevant or not. These heuristics are as follows:

- **Named Entities (NE)**: NE are words that represent the names of persons, locations, organizations, etc. In summarization, the use of this type of elements is widely spread (Conroy et al., 2005; Gupta & Lehal, 2011) and can be helpful when identifying which sentences contain relevant information to construct a summary. In the case of headline generation, NE are important since they can provide information about the persons, the places or the organizations involved in a news article. Therefore, for this heuristic, the NE detected in the three first sentences of the source document are considered as the relevant elements. These NE usually represent the companies, persons or locations in the text. The number of sentences selected for the detection of these elements is not trivial. This is due to the characteristic structure of a news article, where the first few sentences usually contain the most important facts (these sentences commonly answer the 5 W's: Who, What, When, Where and Why). When the corpora is syntactically analyzed and tagged in the preprocessing module of HanaNLG, the information about NE is also retrieved.

- **Latent Dirichlet Allocation** (LDA): In linguistics, a topic or a theme is the subject of the discourse[1], i.e., what the discourse or a text is talking about. The use of topics within summarization is not new and they are usually employed as another way of identifying the sentences that contain information about the main theme of the text when generating a summary (Arora & Ravindran, 2008). In the case of headline generation, they are considered to be the relevant information for generating the headlines since they describe the content of the article. In order to extract these topics, the implementation of LDA generative model in Gensim (Řehůřek & Sojka, 2010) is used. Equation 6.1 shows how this heuristic is calculated in Gensim.

$$P(w,z,\Theta,\beta|\alpha,\eta) = \prod_{i=1}^{k} P(\beta_i|\eta) \prod_{j=1}^{n} P(\Theta_j|\alpha) \prod_{p=1}^{|d_j|} P(z_{p,j}|\Theta_j) P(w_{p,j}|\beta_{z_{p,j}}) \tag{6.1}$$

where w is a word contained in the corpus, z is the topic indicators of each corpus word, Θ is the topic-by-documents distribution , β is the word-by-topic distribution, α (resp. η) are priors on the document mixtures, k is the number of topics, n the total number of words in all documents and $|d_j|$ denotes the length of the document j in words.

- **Term Frequency-Inverse Sentence Frequency** (TF-ISF): There are some widely used statistics that help to determine whether a word is important within a document or not. One of these statistics is TF-ISF, which was first implemented as an adaptation from document retrieval to sentence retrieval (Zhang et al., 2004). It is a numerical statistic which indicates the importance of a word within a specific sentence in a document. This heuristic is calculated following the formula shown in Equation 6.2.

$$tf - isf_{t,s} = f_{t,s} \cdot \log \frac{N}{n_t} \tag{6.2}$$

where, $f_{t,s}$ is the number of occurrences of term t in the sentence s, N is the total number of sentences in the document and n_t is the number of sentences that contains the term t.
Although TF-ISF is similar to Term Frequency-Inverse Document Frequency

[1] https://www.merriam-webster.com/dictionary/topic

(TF-IDF), the latter is not appropriate to this task since this heuristic is used when dealing with more than one document. As we are working with single-document summaries, the former is more suitable. When calculating this statistic, using a threshold allows us to determine whether or not a word is important with respect to each of the sentences within the document.

- **Term Frequency** (TF): Another widely used statistic in the NLP research area is TF. This numerical statistic indicates the relevance of a term within a document, based on its frequency. Moreover, it has been shown that the more frequent a word is in a text, the more probable is its appearance in the final summary (Nenkova et al., 2006). The formula to compute it is shown in Equation 6.3.

$$tf_{t,d} = f_{t,d} \tag{6.3}$$

where, $f_{t,d}$ is the frequency of a certain term in a document, i.e., the number of times that the term f appears in document d.

These heuristics are used in the *vocabulary selection* module within the architecture of HanaNLG (see Section 4.2 of Chapter 4). Therefore, the list of relevant elements (i.e., words or NEs) obtained from these heuristics will be stored in the bag of words employed during the generation.

The remaining modules of HanaNLG do not need to be changed or adapted for the generation of headlines, so that its performance is the same as that described in the Chapter 4. For generating the headlines, “lemmas" were used as the level of abstraction for HanaNLG, since they were the ones which yielded the best results on the evaluation of our approach (see Section 5.5 of Chapter 5).

6.1.3 Experimental set-up

Several experiments were performed for evaluating the performance of HanaNLG in this task. Specifically, we focused on the task described within the DUC shared task, where the main objective was to create a very short summary, which is comparable to a headline, from a given news article (i.e., single-document summarization). Therefore, the datasets provided in these tasks were used as the input corpora to HanaNLG. The statistics of these corpora are shown in Table 6.1. These datasets were used for training the FLMs used in the generation process as well as for obtaining the words that will be used in the Vocabulary Selection module.

Dataset	Files	Sentences	Avg Sents. per file	Words	Avg Words per file
DUC 2003	624	16,478	27	358,367	575
DUC 2004	500	13,141	27	295,710	592

Table 6.1: Statistics of the documents provided for the task 1 of DUC 2003 and the task 1 of DUC 2004 used during the experimentation.

	Headlines Generated		**Threshold**
	DUC 2003	**DUC 2004**	
NE	624	500	-
LDA	624	500	-
TF-ISF	624	500	0.0075
TF	624	500	0.0005
Total	2496	2000	

Table 6.2: Number of headlines generated in each experiment by the heuristics (i.e., NE, LDA, TF-ISF and TF) and the threshold needed for the generation of the headlines.

Some of the heuristics employed needed a threshold to determine if a word is relevant in a document. In the case of TF-ISF, the threshold 0.0075 is used for the classification of the words within a document as well as to limit the maximum number of words classified. For TF, the threshold 0.0005 is also used to limit the maximum number of words considered as relevant information in the generation process. These thresholds were empirically determined by testing different values and comparing the number of relevant words retrieved.

For each of these heuristics, headlines for every document within the datasets were generated. Thus, resulting in a total of 2,496 and 2,000 headlines (since we have 4 different heuristics) for the DUC 2003 and DUC 2004 datasets, respectively. Table 6.2 summarizes the number of sentences generated using each heuristic and the threshold needed by some of the heuristics.

As far as the tools are concerned, the same tools as the ones detailed in Section 4.7 of Chapter 4 are used. Regarding the configuration of the linear combination of FLM for the ranking module, the same configuration as the one detailed in the previous chapter is employed.

6.1.4 Results and Discussion

In order to assess the headlines generated by HanaNLG two distinct types of evaluation were performed. On the one hand, the generated headlines were manually and automatically evaluated to measure their correctness. On the other hand, the produced headlines were compared to several competitive systems.

Assessing the headlines generated by HanaNLG

- **Manual Evaluation**
  Although the evaluation in summarization is usually performed automatically, we thought that a manual evaluation is worthwhile complement to the automatic evaluation carried out with ROUGE that will be further described. This is because, from a NLG perspective, it is complex to automatically measure the meaning and quality of the generated text. Given the

large number of headlines generated and the impossibility of evaluating them all manually, we decided to assess the headlines corresponding to a representative sample. For extracting this representative sample for the DUC 2003 and DUC 2004 datasets, we used the Formula (6.4) described in (Pita Fernández, 1996), thus resulting in a total of 640 headlines (i.e., 80 headlines for each heuristics and datasets):

$$M = \frac{N * K^2 * P * Q}{E^2 * (N-1) + K^2 * P * Q} \tag{6.4}$$

being N the population, K the confidence interval, P the probability of success, Q the probability of failure and E the error rate. Each value for this parameters was taken as suggested in (Gutiérrez Vázquez et al., 2011):

$$K = 0.95; E = 0.05; P = 0.5; Q = 0.5 \tag{6.5}$$

The main goal of the manual evaluation was to measure the accuracy of the generated headlines according to the following aspects: i) *semantic accuracy* of the produced headline, ii) *grammatical accuracy* and iii) *factual accuracy* (i.e. to what extent does the generated headline reflect the key facts of the news article). These criteria are not based on any existing evaluation criteria; however, we defined them because they can help determine the quality of a generated headline. Therefore, the *semantic accuracy* refers to the degree of semantic meaningfulness. Regarding the *grammatical accuracy*, it involves the correctness of the grammatical structure of the headline. Finally, for the last aspect, *factual accuracy*, the comprehension of the content of the news article is measured based on its headline. In order to assess these aspects, we first performed a user-based collaborative evaluation with 3 assessors. These assessors were graduate and postgraduate students from the Computer Science area with a proficiency level of English. In this regard, several questionnaires, using 5-pt Likert Scale, were designed for the purposes of this evaluation. This type of assessment questionnaires is appropriate and frequently used by the research community (Reiter & Belz, 2009). So, for the first aspect (i.e., *semantic accuracy*), the headlines are scored with the value 1 when the headline is meaningless and 5 otherwise. In the case of *grammatical accuracy*, the value 1 means a lack of grammatical structure and 5 when it is grammatically accurate. Finally, for *factual accuracy*, a headline is scored with the value 1 when it is difficult to understand and 5 when the content of the news articles is inferable from it.

Table 6.3 summarizes the averages of the results obtained for the headlines generated with both datasets. The heuristic which yields better results for both datasets is depicted in the table, is NE. This could be explained by the

	DUC 2003			**DUC 2004**		
Heuristic	***Semantic Acc.***	***Gram. Acc.***	***Factual Acc.***	***Semantic Acc.***	***Gram. Acc.***	***Factual Acc.***
NE	**2.78**	**3.15**	**2.55**	**2.49**	**2.89**	**2.24**
LDA	2.62	3.08	2.29	2.42	2.68	2.10
TF-ISF	2.63	3.11	2.33	2.34	2.63	2.03
TF	2.61	3.14	2.29	2.4	2.68	2.08

Table 6.3: Results of the manual evaluation performed using the DUC 2003 and DUC 2004 datasets for each of the heuristics employed in the *Vocabulary selection* module. These results refer to the averages obtained from the assessors scores.

fact that news articles usually contain information about relevant places, people or organizations that may refer to the news article subject. Since HanaNLG maximizes the number NE appearing during the generation, the resulting headline provides more relevant information.

Figure 6.1 shows the number of headlines produced with the different heuristics for each of the Likert scale values for both datasets. Almost 40% of the headlines generated with the NE heuristic have been scored with the value of 3, in the case of semantic and grammatical accuracy. This indicates that the generated headlines are understandable even though they may not be perfect. Regarding the aspects evaluated, the one with the lowest results is *Factual accuracy*. This could be because the generated headlines may contain many acronyms, interjections and some noun phrases contain many adjectives. This could be solved by expanding the acronyms when generating the headlines and by removing the excess of interjections in the training data. In addition to this, the number of adjectives that a noun phrase can contain can be restricted.

- **Automatic Evaluation**

	DUC 2004			DUC 2003		
System	R1 Recall	R1 Precision	R1 F-Measure	R1 Recall	R1 Precision	R1 F-Measure
NE	**22.62**	27.24	24.52	**23.82**	28.17	25.61
LDA	22.25	26.83	24.15	22.67	26.76	24.35
TF-ISF	20.65	26.01	22.77	23.34	28.68	25.50
TF	22.59	**32.39**	**26.17**	23.81	**33.39**	**27.30**

Table 6.4: ROUGE results, on the DUC 2004 and DUC 2003 datasets, for the headlines generated using the HanaNLG approach. These results refers to the ROUGE-1 recall, precision and F-measure obtained.

Concerning the automatic evaluation, the headlines generated with the four heuristics (i.e. NE, LDA, TF-ISF, TF) were automatically evaluated

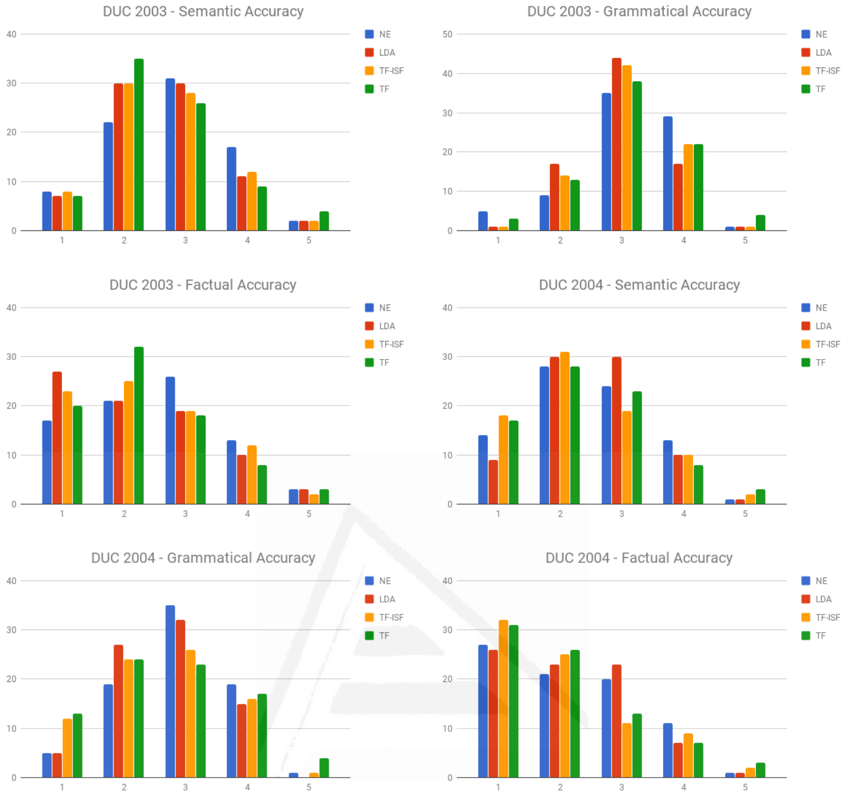


Figure 6.1: Number of sentences scored for each rating of the 5-pt Likert scale regarding the *semantic accuracy*, the *grammatical accuracy* and the *factual accuracy* for both datasets. Axis X represents the rating received for a sentence and axis Y represents the number of sentences that received that rating.

using the metric ROUGE[2] (See Section 3.2.3 of Chapter 3). We selected this tool because it is widely used for evaluating automatically generated summaries. In addition to this, it was employed for automatically assessing the headlines produced in the DUC 2004 share task. ROUGE provides several metrics, including ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4. However, in this evaluation we only used ROUGE-1, which computes the number of coincident unigrams, because the other ones delivered very low results due to the pure abstractive perspective of our method.

Table 6.4 shows the ROUGE 1 results for the produced headlines. In this case, the heuristic NE is also the one which obtained the best results for the recall. As mentioned before, the good results obtained with this heuristic may be due to the presence of locations, people or organizations in

[2] The version 1.5.5 of ROUGE is used, and executed with the following parameters: -a -c 95 -b 75 -m -n 4 -w 1.2

System	**DUC 2003 - Mode**	**DUC 2004 - Mode**
HanaNLG-NE	3	2
HanaNLG-LDA	3	2
HanaNLG-TF-ISF	**1**	4
HanaNLG-TF	4	5
BestDUC03	5	-
BestDUC04	-	**1**

Table 6.5: Results of the manual evaluation performed using the DUC 2003 and DUC 2004 datasets for each of the heuristics compared to the best performing systems of the DUC 2003 and DUC 2004 share task. These results refer to the modes obtained from the assessors scores, where the 1 is the best.

the headlines. However, the recall results may be a bit low because our headlines were generated using the provided datasets whereas the gold-standards were created by humans. Hence, some of the words contained in the gold-standard may differ in some cases from the ones within their respective dataset. Regarding the precision and F-measure, TF was the heuristic which obtained the best results. This heuristic has shown to be a good heuristic for identifying significant terms in automatic summarization (Lloret et al., 2015; Haque et al., 2015).

- **Evaluation with User Preferences Judgements**
  In addition to the manual and automatic evaluation carried out, we also performed user preference judgements (Belz & Kow, 2010) to compare our generated headlines with others. In this sense, we decided to compare our headlines with the ones generated by the following systems:

  - **BestDUC04** (Zajic et al., 2004): The system presented in this paper was the best performing system for the task of producing very short summaries (task 1 - headlines) in the DUC 2004. This system produced extractive summaries.
  - **BestDUC03**: This is the best performing system in the first task of DUC2003.

  For this evaluation, a user-based collaborative evaluation with 3 assessors was conducted. These assessors were the same as those that evaluated the headlines from HanaNLG. In this sense, the assessors were asked to sort in decreasing order the given headlines in order to decide which one is the most and the least preferred. Therefore, the assessors had to sort the provided headlines from 1 (the most preferred one) to 5 (the least preferred one). The results obtained from the assessors, with regard to the mode[3], are shown in Table 6.5.

[3] The value appearing more often in a set of values.

Concerning the headlines generated with the DUC 2003 dataset, the most preferred by the assessors were the ones generated with the TF-ISF heuristic. Alternatively, the ones from BestDUC03 were the least preferred. The reason behind this is that the headlines from BestDUC03 are composed of extracted topics without any type of order while ours are shaped like a complete and understandable sentence.

Regarding the headlines produced using the 2004 dataset, the assessors preferred first the headlines from BestDUC04 and second the ones from HanaNLG, using the NE heuristic. In this case, the headlines from BestDUC04 are most preferred because they are sentences extracted verbatim and contain some topics at the beginning of their headlines.

Comparison with competitive systems

Since the headlines generated with HanaNLG obtained good results in the manual and automatic evaluation, we compared our approach to a set of competitive systems. These systems, which are extractive and abstractive, were specifically developed for headline generation. These systems are as follows:

- **Tan17** (Tan et al., 2017): This system employs a coarse-to-fine approach for generating abstractive summaries. This approach first determines the important sentences of the document and then uses a multi-sentence summarization model with hierarchical attention to construct the headline.

- **Takase16** (Takase et al., 2016): This paper presented an abstractive headline generation model based on an encoder-decoder architecture. This model encodes the results gathered from an abstract meaning representation parser employing a modified Tree-LSTM encoder.

- **Chopra16** (Chopra et al., 2016): This system employs a conditional recurrent neural network to generate a headline by summarizing the input sentence.

- **Rush15** (Rush et al., 2015): This paper presented a method for abstractive sentence summarization. This method uses a local attention-based model to generate a headline conditioned by the input sentence.

- **COMPENDIUM** (Lloret & Palomar, 2013): A text summarization tool which can generate extractive generic language with the use of linguistic and statistical information. For generating the headlines employing this tool, a 5% compression rate was applied to the source documents.

- **LeadBaselineDUC04**: The lead baseline that the participants of the DUC 2004 shared task had to improve. This baseline corresponds to the first 75 bytes of each document.

- **LeadBaselineDUC03**: As in DUC 2004, this was the lead baseline that the participants of the DUC 2003 had to beat. This baseline corresponds to the original headlines of the news articles.

- **BestDUC04** (Zajic et al., 2004): The best performing system in the first task of DUC2004. In order to correctly compare our results, the generated headlines with this system were evaluated using the same version of ROUGE as in our evaluation (i.e. ROUGE version 1.5.5).

- **BestDUC03**: This is the best performing system in the first task of DUC2003. The headlines generated by this system were also evaluated with ROUGE 1.5.5.

Most of these competitive systems produce abstractive summaries, however, they do not use all the information within the source document. For instance, in Tan17, Chopra16 or Rush15; only a small set of the sentences or a solely sentence of the document are employed instead. Subsequently, these headlines are not generated using all the information of the news articles.

The headlines generated with the DUC 2004 dataset were compared to the following systems: Tan17, Takase16, Chopra16, Rush15, Compendium, BestDUC04 and LeadBaselineDUC04. This is due to the fact that they report, in their respective papers, the results for the first task of DUC 2004. Concerning the headlines produced using the DUC 2003 dataset, since most of the competitive systems do not report results for this dataset, they were compare to the headlines created by BestDUC03, LeadBaselineDUC03 and COMPENDIUM systems.

	DUC 2003		
System	**R1 Recall**	**R1 Precision**	**R1 F-Measure**
COMPENDIUM	13.71	10.56	11.84
BestDUC03	**28.27**	**29.96**	**28.85**
LeadBaselineDUC03	19.18	25.04	21.31
HanaNLG-NE	23.82	28.17	25.61
HanaNLG-TF-ISF	23.34	28.68	25.50

Table 6.6: ROUGE results on the DUC 2003 dataset for the competitive systems and the *HanaNLG* approach with the NE heuristic. These results refers to the ROUGE-1 recall, precision and F-measure obtained.

The ROUGE results of our generated headlines using the heuristic which delivered the best results (i.e. NE) in comparison to the competitive systems ones are shown in Table 6.6 and Table 6.7. The results in the case of Tan17, Takase16, Chopra16 and Rush 15 were directly extracted from their respective papers. For the systems and lead baselines from the DUC 2003 and DUC 2004 shared tasks — BestDUC03, BestDUC04, LeadBaselineDUC03 and LeadBaselineDUC04 — , the ROUGE results were recalculated employing ROUGE 1.5.5. Since their generated

	DUC 2004		
System	**R1 Recall**	**R1 Precision**	**R1 F-Measure**
Tan17	**28.97**	-	-
Takase16	28.80	-	-
Chopra16	28.68	-	-
Rush15	26.55	-	-
COMPENDIUM	14.08	12.50	13.12
BestDUC04	25.65	**27.36**	**26.26**
LeadBaselineDUC04	22.25	23.74	22.83
HanaNLG-NE	22.62	27.24	24.52
HanaNLG-TF-ISF	20.65	26.01	22.77

Table 6.7: ROUGE results on the DUC 2004 dataset for the competitive systems and the *HanaNLG* approach with the NE heuristic. These results refers to the ROUGE-1 recall, precision and F-measure obtained.

headlines were publicly available, this recalculation was done in order to evaluate the system under the same conditions as HanaNLG.

Although our approach does not outperform the best competitive systems, our approach was found to deliver comparable results, taking into account that it was not firstly designed as a summarization system. With respect to the lead baselines defined for each of the DUC tasks, HanaNLG surpasses their results. In addition to this, HanaNLG also achieved better results than some of the DUC 2003 and DUC 2004 participants. In the case of participating in the first tasks of DUC 2003 and DUC 2004, HanaNLG would have ranked 2nd among the 13 DUC 2003 systems (see Table 6.8) and would have also ranked 2nd among the 18 DUC 2004 systems (see Table 6.9). Moreover, we obtain better results than the COMPENDIUM system. Although COMPENDIUM is an extractive and generic summarization system, extracting the most important sentence of the document is not enough to be considered as a headline. Therefore, this type of system is not suitable for this task.

		DUC 2003		
Rank	**System**	**R1 Recall**	**R1 Precision**	**R1 F-Measure**
1	BestDUC03	28.27	29.96	28.85
2	HanaNLG-NE	23.82	28.17	25.61
3	SSID-17	23.12	20.71	21.62
4	SSID-21	21.66	21.78	21.61
5	SSID-18	20.28	19.76	19.77

Table 6.8: Ranking of the task 1 of DUC 2003 if HanaNLG had participated in the shared task.

Some examples of the sentences generated using the DUC 2003 and DUC

		DUC 2004		
Rank	**System**	**R1 Recall**	**R1 Precision**	**R1 F-Measure**
1	BestDUC04	25.65	27.36	26.26
2	HanaNLG-NE	22.62	27.24	22.52
3	LeadBaselineDUC04	22.25	23.74	22.83
4	SSID-77	22.24	24.17	22.33
5	SSID-131	22.16	24.85	23.11

Table 6.9: Ranking of the task 1 of DUC 20014 if HanaNLG had participated in the shared task.

2004 are shown in Example 20.

(20) **DUC 2003**

Military Mubarak in Turkey eradicates unannounced Arab in Syria.

The new opening of the terrorism gathers international Interpol of strategy.

DUC 2004

The political head in Congo revitalizes the eastern people in Rwanda.

The western rebel in Congo bolsters the weary base in Kinshasa.

To sum up, the proposed NLG approach, HanaNLG, is capable of generating sentences guided by the topics or themes extracted from a news articles. This leads to a sentence that can be considered as a headline since this sentence contains information related to specific topics. Despite not overcoming the results from competitive summarization systems, this approach obtained promising results in manual and automatic evaluations. Furthermore, our results improve on the ones delivered by other summarization-focused systems.

6.2 HanaNLG for Cross-document Timeline Generation

In Section 6.1 we used HanaNLG for the generation of headlines, a task that can be considered as a single-document summarization task. In this section we will apply HanaNLG for generating a narrative multi-document summary that describes the events that occurred in relation to a concrete entity in the form of a narrative timeline.

Given the enormous amount of information available nowadays, text summarization is key in terms of providing mechanisms to automatically summarize the information contained in different documents (Mani, 1999). The optimal structure to present this information is the narrative structure (Gottschall, 2012), where different events about one or more entities are arranged following a time order. Each of these events is a fact occurring in the text at a specific moment with a specific structure (Hovav et al., 2010). These events can denote activities, states or accomplishments (Mani et al., 2005) and may involve several participants or components (such as time, place, etc.) (Ji et al., 2009).

The main objective of this experimentation is to assess the adaptability of HanaNLG to produce multi-document summaries. In this sense, as mentioned before, our goal is to generate narrative summaries based on a natural time ordering of events (a timeline) from a set of source documents. This task is called cross-document timeline generation. The timeline collects the events of a specific entity that appear in various documents and presents them in an orderly manner. The use of NLG techniques in the generation of this type of summary could be useful since they could provide more flexibility in the way that the summary is produced.

In this section, we describe the adaptation of HanaNLG to cross-document timeline generation and the evaluation carried out. Therefore, we first define the task to perform (Subsection 6.2.1), and then, we detail the adaptation of HanaNLG for generating cross-documents timelines (Subsection 6.2.2). Finally, we jointly describe the experimentation and evaluation conducted (Subsections 6.2.3 and 6.2.4).

6.2.1 Task Description

The task of cross-document timeline generation is twofold: i) cross-document timeline extraction and ii) summarization. On the one hand, cross-document timeline extraction involves the identification, obtaining and chronologically ordering the events related to a target entity (Minard et al., 2015). On the other hand, summarization is responsible for generating a chronological ordered summary from the events extracted. As far as our work is concerned, the latter task is closely related to NLG.

The generation of these summaries can be performed from the two perspectives aforementioned: extractive summarization and abstractive summarization. However, the former may only extract the sentences involved with an event and put them in the summary verbatim, losing the temporal connections that appear in the text. Therefore, the latter is more appropriate for these cases, where the information from the events and their temporality can be expressed as desired, and therefore, that is why we apply HanaNLG to integrate NLG with summarization and to obtain abstractive summaries.

6.2.2 Adapting HanaNLG for Cross-document Timeline Generation

In order to be able to generate abstractive summaries containing temporal references and events of a given target entity it is essential to first extract this kind of information. In this sense, if HanaNLG were a full NLG system (i.e., including all the stages mentioned: macroplanning, microplanning and surface realization) this type of information would be identified and obtained during the macroplanning. Since this thesis only tackles the surface realization stage, we need to rely on other resources to obtain this temporal information and events relating to a target entity. The resources used for obtaining this temporal information as well as the adaptation of HanaNLG will be described below.

Extraction of Temporal Information and Events

We make use of an Enriched Timeline Extraction system, which is an improved version of the one presented in (Navarro-Colorado & Saquete, 2016), to obtain the temporal information and events. The output of this system, which will form part of the input of HanaNLG, is an enriched timeline containing a set of enriched clusters of events as the one shown in Example 21. These clusters contain the main event (which is usually a verb) and a set of arguments (i.e., the arguments may refer to different semantic roles: A0, A1, A2, A3 and A4) related to that event.

(21) 0 2008 en-82548-4-built:$_{(A1,The\ plane),(A2,with\ four\ Rolls-Royce_{T}rent\ 900\ engines)}$

(EN: In 2008, they built the plane with four Rolls-Royce_Trent 900 engines)

en-82548-2-made:$_{(A1,The\ first\ A380\ superjumbo),(A0,by\ Airbus)}$

(EN: In 2008, Airbus made the first A380 superjumbo)

Adaptation of the Generation Process

For each of the cluster of events from the enriched timeline, a sentence, representing the information related to a specific event, will be generated by HanaNLG. Therefore, to be able to generate narrative summaries with the information provided by the enriched timelines, some of the modules within HanaNLG were adapted. In this sense, in the preprocessing module, we first check, for each of the events contained in the enriched timeline, if there are duplicate semantic roles in the same event. In this case, if two or more arguments refer to the same semantic role in a specific event, we will choose the most probable argument to be used during the generation. This argument is selected based on the probability of the phrase contained in it, which is calculated by the ranking module of HanaNLG.

Once the duplicate semantic roles are removed from the events of the enriched timeline, the content of the arguments of each event (the words within these arguments) is used as the seed feature (in this case we use the type of seed feature“*keyword*"). This content will constitute the vocabulary of the summary that will be generated. Then, for each cluster of events of the enriched timeline a sentence is generated following the procedure previously explained (see Section 4.3 of Chapter 4) taking into account some aspects. The components of the frames used during the generation may need some type of particular semantic role, such in the case of an agent (i.e. A0, A1) or and instrument (i.e. A2). In these cases, only the content of that specific argument will be used for generating that specific component of the sentence. For example, in the case that the frame indicates that the verb needs a Subject, and, the event has an argument *ARG* with the semantic role A0, the content of the argument *ARG* will be used to form the Subject of the sentence.

Finally, the remaining modules of our approach do not need to be adapted for generating this type of summary. Therefore, its performance is the same as the one previously described in Chapter 4.

6.2.3 Experimental set-up

In order to evaluate the summaries generated by HanaNLG several experiments were performed. In this regard, the test dataset from the Task 4 at SemEval 2015[4] was used as the corpus in this experimentation. This dataset is composed of 90 documents from Wikinews articles in English about diverse topics such as Airbus, General Motors or Stock Market. Table 6.10 summarizes the main statistics of this dataset.

# of documents	# of target entities	# of tokens	# of events
90	35	≈30,000	915

Table 6.10: Statistics of the test dataset provided in the Task 4 of SemEval 2015

Each of the documents of this dataset may contain information about several target entities. Therefore, for each of the entities provided in the dataset, a narrative abstractive summary composed only with events related to the target entity will be generated in this experimentation. These summaries were generated considering two configurations: (i) gold-standard experiment and (ii) overall system experiment. Hence, a total of 70 narrative summaries — 35 summaries for each experiment — were produced.

Concerning the gold-standard experiment, we used the gold-standard timelines provided in the Task 4 of SemEval. These timelines can be used for assessing the summaries generated by HanaNLG. Thus, avoiding the errors derived from the enriched timeline generated.

In the case of the overall system experiment, we used unlabeled data for evaluating the approach in a real scenario. In this sense, the Enriched Timeline Extraction system was used to extract an intermediate timeline schema from the raw data of the SemEval corpus. This scheme contains the event and the temporal information that HanaNLG will be used to compose the final summary.

In addition, for comparing the results in these experiments, several extractive summarization systems were used. In this regard, the following systems were employed: COMPENDIUM (Lloret & Palomar, 2013), GRAFENO (Sevilla, Fernandez-Isabel, & Díaz, 2016) and Open Text Summarizer (OTS) (Andonov, Slavova, & Petrov, 2016). These systems allow the generation of multi-document summaries; however, to be able to generate entity-focused extractive summaries

[4] http://alt.qcri.org/semeval2015/task4/index.php?id=data

with them, the input documents were preprocessed. In this sense, all the documents belonging to the same corpus (i.e., that are related to the same target entity) were merged into a single document. Then, from the merged document, the sentences that are not strictly related to the target entity were removed. In this way, the extractive summarization using these systems was focused on determining the important information related to an entity. A total of 35 summaries were generated by each of the aforementioned systems.

Finally, we also defined two distinct baselines for narrative abstractive summarization: *FirstEvent* and *LongestEvent*. On the one hand, for the *FirstEvent* baseline, a summary was generated using the first event of the cluster of events. On the other hand, in the case of the *LongestEvent* baseline, a summary employing the event with the highest number of arguments was produced. Each of these baselines were generated for both experiments: (i) the gold-standard and (ii) the overall system experiments.

6.2.4 Evaluation and results

For assessing the generated summaries, we carried out two different types of evaluations, as in the case of the generation of headlines. In this regard, we first assessed the summaries generated manually and automatically and compared these results with those of the baselines and extractive summarization systems. Then, we evaluated the summaries generated in the context of timeline summarization and compared our results to the results of some competitive systems.

Assessing the summaries generated by HanaNLG

As previously mentioned, we assessed the generated summaries manually and automatically. In this sense, for the manual evaluation we conducted a user-based evaluation from a readability perspective. A total of 12 assessors, with proficient level of English, participated in this evaluation. The assessors were asked to answer a questionnaire[5] that tackled the readability and linguistic criteria defined for the tracks of the DUC and TAC conferences. Specifically, we evaluated the *summary's grammaticality*, *non-redundancy*, *referential clarity*, *focus*, as well as *structure* and *coherence*. Additionally, the *overall responsiveness* of the summary was evaluated to determine to what extent the summary satisfied the task requirement.

In this evaluation we compare the summaries generated by HanaNLG with the ones from the previously mentioned baselines — FirstEvent and LongestEvent —. The reason behind this is that the summaries of HanaNLG as well as these baselines were generated employing NLG techniques.

The average results of this manual evaluation, for both experiments — (i) gold-standard experiment and (ii) overall experiment —, are shown in Table 6.11

[5] https://goo.gl/buC68B

and Table 6.12 respectively.

	Readability/Fluency						Overall
	Grammaticality	Non-redundancy	Referential clarity	Focus	Structure and Coherence	Average	Responsiveness
FistEvent	2.47	2.70	2.73	2.42	1.97	2.46	2.16
LongestEvent	2.08	2.77	2.80	2.30	1.85	2.36	2.03
HanaNLG	**2.78**	**3.18**	**3.36**	**3.25**	**2.83**	**3.08**	**2.89**

Table 6.11: Average values for readability/fluency (including the average values for summary's grammaticality, non-redundancy, referential clarity, focus and structure and coherence) and for the summary's overall responsiveness for the (i) gold-standard experiment.

	Readability/Fluency						Overall
	Grammaticality	Non-redundancy	Referential clarity	Focus	Structure and Coherence	Average	Responsiveness
FistEvent	2.52	2.81	2.84	3.00	2.33	2.70	2.74
LongestEvent	2.45	2.76	3.05	2.90	2.21	2.67	2.66
HanaNLG	**2.69**	**3.41**	**3.53**	**3.79**	**3.07**	**3.30**	**3.60**

Table 6.12: Average values for readability/fluency (including the average values for summary's grammaticality, non-redundancy, referential clarity, focus and structure and coherence) and for the summary's overall responsiveness for the (ii) the overall system experiment.

HanaNLG obtained better results than the baselines in both experiments as seen in the Tables. Indeed, the summaries generated by our approach surpass the results obtained by the baselines for each of the parameters evaluated. This indicates that the linguistic quality of the produced text increases when using HanaNLG in comparison to the baselines defined.

Furthermore, we also carried out a user preference judgement (Belz & Kow, 2010) where the assessors had to order the summaries according to their preferences. In this regard, the assessors preferred the summaries generated by HanaNLG in both experiments. Specifically, they preferred in 79.45% and 79.66% of the cases our summaries in the gold-standard experiment and the overall experiment, respectively.

Concerning the automatic evaluation, the tool ROUGE (version 1.5.5) was used for evaluating how informative a summary is by comparing its content with reference documents. Therefore, in order to be able to assess the summaries automatically employing ROUGE, reference summaries are needed. In this regard, we created reference summaries directly from the gold-standard timelines, available in the corpus used, using a semi-automatic process. For generating each sentence of these summaries, the following steps are followed:

- *Verb selection*: We select the first verb in the cluster of events for the creation of the sentence.

- *Arguments selection*: For producing the sentence, we only select one argument for each type of argument (i.e., A0, A1, A2, A3 or A4). In the case of having more than one argument for one concrete type, the one with the longest length is chosen. This is because it may contain more information about the target entity.

- *Sentence generation*: The sentences, for each of the clusters, are generated using a pattern:

  (22) **Pattern**: *Time* A0 *event* A1 A2 A3 A4

  We use the arguments available in each case. Since A0 and A1 are essential arguments, in the case of one not including an essential argument, the target entity is used instead.

  In case of nominalizations, since it is not possible to obtain semantic roles from them, the sentence is produced employing the following pattern:

  (23) **Pattern**: *Time TargetEntity* had a *NominalizationEvent*

For this automatic evaluation, we compare our ROUGE results with the ROUGE results from the extractive systems — COMPENDIUM, GRAFENO and OTS — and the baselines — FirstEvent and LongestEvent —. Tables 6.13 and 6.14 show the average results for ROUGE recall (R), precision (P) and F1-measure (F) for the following metrics: (i) ROUGE-1, (ii) ROUGE-2, (iii) ROUGE-L and (iv) ROUGE-SU4. Where ROUGE-1 and ROUGE-2 computes the number of coincident unigrams and bigrams; ROUGE-L calculates the longest matching subsequence with a reference summary; and ROUGE-SU4 measures the matching skip-bigrams with a maximum distance of four words.

	ROUGE-1			**ROUGE-2**			**ROUGE-L**			**ROUGE-SU4**		
	R	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
COMPENDIUM	0.317	0.370	0.312	0.114	0.154	0.121	0.296	0.348	0.293	0.142	0.180	0.145
GRAFENO	0.285	0.415	0.295	0.102	0.199	0.118	0.261	0.384	0.272	0.127	0.140	0.139
OTS	0.305	0.362	0.303	0.106	0.148	0.114	0.280	0.335	0.280	0.133	0.173	0.138
FirstEvent	0.323	0.583	0.402	0.141	0.270	0.179	0.316	0.570	0.392	0.140	0.264	0.176
LongestEvent	0.351	0.688	0.445	0.166	0.335	0.215	0.340	0.665	0.431	0.165	0.339	0.214
HanaNLG	**0.576**	**0.735**	**0.637**	**0.420**	**0.544**	**0.467**	**0.559**	**0.714**	**0.619**	**0.400**	**0.518**	**0.445**

Table 6.13: Average values for recall, precision and F1-measure for the gold-standard annotations ((i) gold-standard experiment). Comparison between different summarization and baseline approaches.

As seen in the tables, in both experiments, we outperformed the remaining systems. This indicates that the combination of NLG techniques with the identification of events and extraction of temporal information enhances narrative summarization. However, the results obtained for the overall experiment are lower than the ones from the gold-standard experiment. This is due to the fact that the Enriched Timeline Extraction system may introduce errors when extracting the events of the documents.

	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4		
	R	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
COMPENDIUM	0.317	0.370	0.312	0.114	0.154	0.121	0.296	0.348	0.293	0.142	0.180	0.145
GRAFENO	0.285	0.415	0.295	0.102	0.199	0.118	0.261	0.384	0.272	0.127	0.140	0.139
OTS	0.305	0.362	0.303	0.106	0.148	0.114	0.280	0.335	0.280	0.133	0.173	0.138
FirstEvent	0.258	0.463	0.302	0.083	0.164	0.101	0.250	0.444	0.293	0.100	0.194	0.119
LongestEvent	0.251	0.524	0.312	0.088	0.196	0.114	0.245	0.510	0.305	0.099	0.225	0.125
HanaNLG	**0.433**	**0.595**	**0.470**	**0.263**	**0.363**	**0.284**	**0.422**	**0.579**	**0.457**	**0.260**	**0.360**	**0.282**

Table 6.14: Average values for recall, precision and F1-measure when using raw data without any type of annotation as input ((ii) overall system experiment). Comparison between different summarization and baseline approaches in a real scenario.

	COMPENDIUM				GRAFENO				OTS				FirstEvent				LongestEvent			
	R1	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**	**R1**	**R2**	**RL**	**RSU4**
HanaNLG	105	286	111	207	116	295	128	220	110	309	121	223	59	160	58	153	43	117	43	108

Table 6.15: Percentage of improvement for the F1-measure metric when comparing HanaNLG with respect to the extractive summarization approaches and abstractive baselines for the gold-standard annotations ((i) gold-standard experiment). R1, R2, RL, and RSU4 refer to ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4, respectively.

Tables 6.15 and 6.16 report the percentage of improvement obtained by HanaNLG with respect to the extractive systems and the defined baselines. For calculating this improvement, we only considered the F1-measure values. The results reported in the tables indicate that the performance of HanaNLG is better compared with the other summarization approaches in both experiments. In addition to this, although one of the baselines (i.e., LongestEvent) is more competitive than the other (i.e., FirstEvent), our approach is capable of obtaining better results.

Example 24 shows a fragment of a narrative summary generated by HanaNLG about the target entity "Ford".

(24) 2006-01:The company beat last year 's profit of $ 11 billion .

2006-01-23:Ford committed to invest $ 200 million into the plant to upgrade the appearance of the two cars manufactured there .

2006:The company reported a second quarter loss of $ 254 million and a 34% year-year decline in sales for the month of July .

2009:Ford had a $ 34.3 billion debt .

	COMPENDIUM				GRAFENO				OTS				FirstEvent				LongestEvent			
	R1	R2	RL	RSU4	R1	R2	RL	RSU4	R1	R2	RL	RSU4	R1	R2	RL	RSU4	R1	R2	RL	RSU4
HanaNLG	51	135	56	95	59	140	68	103	55	149	65	105	56	182	56	137	51	153	50	125

Table 6.16: Percentage of improvement for the F1-measure metric when comparing HanaNLG with respect to the extractive summarization approaches and abstractive baselines when using raw data without any type of annotation as input ((ii) overall system experiment). R1, R2, RL, and RSU4 refer to ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-SU4, respectively.

Comparison with the state of the art

In addition to the experimentation carried out to evaluate the generated language, we conducted an evaluation in the context of timeline summarization. To the best of our knowledge, there are no specific datasets with reference summaries for the task of timeline generation. Therefore, since our approach obtained good results, we also wanted to validate its performance with datasets created for a similar task (i.e., timeline summarization). The main difference between the summaries generated for the task of timeline summarization and our generated summaries is that ours aim to be narrative instead of only timelines. In contrast, the task of timeline summarization aims to generate a timeline summarizing the information of one or more input documents. These timelines contain short summaries that are ordered by the document creation time.

The dataset chosen for this evaluation is Timeline17, which has been used by different timeline summarization systems (Tran et al., 2013; Binh Tran et al., 2013) and it is available online[6]. This dataset is composed of a total of 4,650 news articles automatically gathered from diverse sources (e.g., CNN, BBC or NBCnews) about 9 topics.

In order to be able to produce the summaries using this dataset, we considered the topics to be the target entities for constructing the summaries. The Enriched Timeline Extraction system was first employed to extract the event and temporal information. Then, HanaNLG was used for generating the narrative summaries HanaNLG from the output provided by the Enriched Timeline Extraction system.

We used ROUGE to automatically assess the generated summaries with respect to the reference summaries provided in the dataset. For assessing the summaries under the same conditions some aspects were considered. In this regard, ROUGE was set to truncate the length of the summaries to the same length as that of the reference summaries.

Table 6.17 shows the average F1-measure results for the ROUGE metrics evaluated in contrast to several timeline summarization systems. In this regard, we only reported the F1-measure for ROUGE-1, ROUGE-2, and ROUGE-SU4 because they were the values reported by the timeline summarization systems

[6] http://www.l3s.de/~gtran/timeline/

	ROUGE-1	ROUGE-2	ROUGE-SU4
	F	F	F
COMPENDIUM (Lloret & Palomar, 2013)	0.340	0.085	0.133
GRAFENO (Sevilla et al., 2016)	0.267	0.069	0.102
OTS (Andonov et al., 2016)	0.337	0.076	0.127
Chieu et al.(Chieu & Lee, 2004)	0.202	0.037	0.041
MEAD(Radev et al., 2004)	0.208	0.049	0.039
ETS(Yan et al., 2011)	0.207	0.047	0.042
Tran Linear Regression(Binh Tran et al., 2013)	0.218	0.050	0.046
Tran LTR(Tran et al., 2013)	0.230	0.053	0.050
HanaNLG	**0.413**	**0.121**	**0.176**

Table 6.17: Average F1-measure values when using Timeline17 dataset as input. Comparison between different multi-document and timeline summarization approaches.

in their respective papers. In addition to these systems, we also compared our results with the ones obtained by the extractive systems previously described (i.e., COMPENDIUM, GRAFENO and OTS). As seen in the table, our results outperform both, the timeline summarization and the extractive systems. This could be due to the fact that, besides using an enriched timeline as input to HanaNLG, we exploit the temporal information, links and expressions related to the target entity events to produce the summary.

6.3 Conclusion

This chapter presented the usefulness and applicability of HanaNLG for several tasks related to automatic summarization. Specifically, these tasks were: (i) headline generation and (ii) cross-document timeline generation. In light of the results obtained, the adaptability of HanaNLG for different domains and purposes has been proven. Concerning the first task of headline generation, although our approach did not outperform the results from the competitive systems presented, they were better than other summarization focused systems, such as COMPENDIUM or the lead baselines of the DUC share task (i.e., LeadBaselineDUC03 and LeadBaselineDUC04). Moreover, if our approach had participated in the DUC 2003 and DUC 2004 shared tasks we would have ranked 2^{nd} in both tasks. In the case of the second task, cross-document timeline generation, our approach has shown promising results with different datasets, outperforming the results of several summarization systems. Therefore, the main contributions of this chapter can viewed from two perspectives:

- **Analysis of the adaptability and usefulness of HanaNLG to different purposes and domains within other NLP areas.** The experimentation conducted has proven the adaptability of our approach to different applications within the text summarization field with good results. For instance, in the case of headline generation, we generated single-document abstractive headlines summarizing the gist of the document. Regarding multi-document summarization, in the cross-document timeline generation task, the use of a Enriched Timeline Extraction system in conjunction with HanaNLG allows the generation of narrative abstractive timelines. In both cases, the use of NLG techniques allows the generation of abstractive summaries.

- **Extrinsic evaluation of the language generated by HanaNLG.** In addition to the adaptation of our approach to diverse NLP applications, HanaNLG is being evaluated extrinsically to provide an idea of its performance in these applications (i.e., headline generation and cross-document timeline generation). This type of evaluation complements the intrinsic evaluation conducted in Chapter 5, thus verifying the appropriateness of the proposed approach for generating language that is easily adaptable for different domains and purposes. In addition to this, the integration of HanaNLG does not harm the results obtained but, on the contrary, improves them as in the case of cross-document timeline generation.

CHAPTER 7

Conclusion and Work in Progress

From its beginnings in the 1950s, the methodologies and systems within the Natural Language Generation (NLG) field have experienced a breakthrough. New approaches have been developed, taking into account the type of input to the system (e.g., T2T and D2T) or the target for which the system was created (e.g., generation of informative texts, generation of persuasive texts or dialogue systems). In addition, different types of approaches (e.g., knowledge-based, statistical or deep learning approaches) have been designed for tackling the tasks associated with NLG (e.g., content selection, sentence aggregation or linguistic realization). However, there is still a lot of room for improvement, especially due to the way that the existing systems are developed. In this sense, NLG systems are currently designed for very specific domains, purposes and languages. Therefore, research on flexible and independent approaches for generating multi-purpose texts is an open challenge in this arena. The evaluation of NLG systems and their outputs is also an open challenge. The reason behind this is the fact that there is no consensus in the research community on how to evaluate these systems, whether automatically or manually.

Within this context, this research thesis has addressed the NLG task, providing first information about the state-of-the-art and how the systems are evaluated. In Chapter 2, we performed an extensive analysis of how the NLG systems can be classified and we described the different tasks that compose the architecture of a NLG system with the purpose of knowing how the current NLG systems are developed. Moreover, the approaches employed to address the task of NLG were also detailed. Chapter 3 provided information about the existing resources for NLG as well as the most common methodologies followed for the evaluation of this type of system. Then, from the conclusions extracted in these two state-of-the-art chapters, where the need for research into more flexible NLG methods is evident, we proposed HanaNLG in Chapter 4. HanaNLG is a hybrid NLG approach capable of flexibly adapting the generated text/content to different domains, scenarios and purposes. This is done through the combination of knowledge-based resources and statistical information together with the use of seed features. These seed features are considered to be abstract objects (e.g., phonemes, polarities, topics, etc.) that are used to guide the generation process in terms of content. Regarding the generation of text itself, we employ VerbNet and WordNet as lexical resources and FLM as the statistical model to produce text. HanaNLG takes as input the seed feature previously mentioned, a corpus, the number of sentences to generate, the level of abstraction (i.e., a variable that indicates that the sentence will be generated using words, lemmas or synsets), a variable indicating the generation of related sentences and the verb tenses of the generated sentences. As a result of the generation process performed by HanaNLG, an inflected and well-constructed text is obtained. With respect to HanaNLG's architecture, it is comprised of six different modules: (i) *preprocessing*; (ii) *vocabulary selection*; (iii) *sentence generation*; (iv) *sentence ranking*; (v) *sentence inflection* and (vi) *sentence aggregation*. The intrinsic assessment of HanaNLG (i.e., with respect to the content and the quality of the generated text) in several scenarios (Chapter 5) and its extrinsic assessment by analyzing its adaptability to automatic summarization tasks (Chapter 6), has proven to yield good results in the generation of texts for different purposes and scenarios.

The remaining sections will summarize the main contributions (Section 7.1), the work in progress as well as future work (Section 7.2) and the list of relevant publications related to this thesis (Section 7.3).

7.1 Main Contributions

Summing up all the research work carried out in this thesis, the main contributions are:

- **Analysis of the state-of-the-art concerning the approaches and methodologies for generating natural language**. From the extensive analysis carried out, we were able to discern some insights on future directions of the NLG field. This is related to the need for designing and developing more flexible and versatile approaches. In this sense, since the existing systems are currently designed for concrete purposes, domains and languages, their adaptation to other domains is usually very costly. Therefore, research on more easily adaptable and flexible approaches would be a major step forward in this research area. Regarding the approaches used for developing the NLG systems, the most common are the ones that use knowledge-based techniques but also the ones employing statistical techniques. The use of approaches employing deep learning techniques is increasing; however, they have not been sufficiently tested and the text generated by some of them may contain incorrect information.

- **Analysis of the state-of-the-art in the evaluation of generated language**. Despite the existing evaluation methodologies, either automatic or manual, the evaluation of the text automatically generated by NLG approaches is still an open challenge. When considering the automatic evaluation in NLG, this is conducted using metrics from other NLP areas. However, these types of metrics cannot guarantee that a text is meaningful or that its grammatical structure is fully correct. In addition to this, since in NLG there is no single correct output and there is a lack of gold-standards, this makes the manual evaluation of the texts preferable in many cases. This manual evaluation is usually performed collaboratively and through the use of questionnaires. However, the main concern of this type of evaluation is the subjectivity of the assessors, leading to a high variance in the judgements. Regardless of the type of evaluation methodology, there is no consensus on which one to use, so researchers decide according to their needs.

- **Proposal and development of HanaNLG**. In light of the limitations detected through the analysis of the state-of-the-art of NLG, we designed and developed HanaNLG, which is a hybrid approach based on the surface realization stage. This approach is capable of easily adapting the language generated to different scenarios and domains. This is done through the use of seed features in conjunction with hybrid techniques (i.e., combination of knowledge-based information — VerbNet and WordNet — and statistical models — FLM —). The architecture of HanaNLG is composed of six different modules. The *Preprocessing* and the *Vocabulary selection* deal with the preprocessing of the input, the training of the FLM used during the generation process and the selection of the vocabulary that will compound the final text. The last four modules — *Sentence generation*, *Sentence ranking*, *Sentence inflection* and *Sentence aggregation* — are responsible for the generation of the final output of the approach. This output will be a sentence or set of inflected and well structured sentences.

- **Intrinsic evaluation of HanaNLG** In order to verify the appropriateness of the techniques used during the development of HanaNLG as well as the complete approach, we conducted an incremental evaluation, which allowed the assessment of every individual aspect in the development of HanaNLG. First, analyzing the suitability of the language models employed and the use of seed features in the generation of language. Then, analyzing the flexibility of adaptation for the generation in different scenarios. And, finally, assessing HanaNLG as a whole and complete hybrid surface realization approach. In this evaluation, we obtained good results in the generation of sentences that are easily adaptable to different scenarios and purposes. That is, HanaNLG was tested in two scenarios: the NLG for assistive technologies scenario; and, the NLG for opinionated sentences scenarios, respectively delivering results of 97.73% and the 100% with respect to generating sentences that were meaningful and that included seed features. Moreover, the quality of the language generated has improved due to the inclusion of an inflection module within our approach and the combination of semantic resources and statistical models. In this sense, the number of meaningful generated sentences has increased as well as the number of meaningful newly generated sentences including seed features, with respect to the version of HanaNLG where semantic information was not used. Furthermore, some conclusions about the advantages and limitations of HanaNLG can be drawn from the extensive evaluation carried out. As for the advantages, HanaNLG has proven to be capable of easily adapting the language generated to different domains or purposes thanks to the use of seed features. In addition to this, the combination of semantic resources and statistical information in conjunction with the seed features has shown to increase the flexibility of the language generated in terms of vocabulary. In contrast, its main limitation is twofold. Firstly, the sentences generated are short due to the structure obtained from the frames used during the generation process. Secondly, some of the resources employed in the development of our approach are language dependent. Therefore, other resources for a specific target language may be needed in order to be able to generate for that language. These issues could form part of the future research directions which could benefit and improve HanaNLG.

- **Extrinsic evaluation of HanaNLG as an application for the task of automatic summarization**. Since the language generated by HanaNLG has yielded good results in different scenarios, it is important to analyze the adaptability of this approach to other NLP fields. In particular, we focused this analysis on the automatic summarization area. Within this area, we tested HanaNLG under two different applications: headline generation and cross-document timeline generation. Within the first application, HanaNLG was used to generate single-document abstractive summaries that summarize the main idea of a news article in the form of a headline. In the case of the second application, we generated multi-document timeline summaries with the use of an external Enriched Timeline Extraction module integrated on top of HanaNLG. These summaries contain the events of a specific entity appearing in several documents and are presented in an orderly manner. Finally, from the results of both applications, it is worth mentioning that although the results obtained by the adaptation of HanaNLG for the task of headline generation do not outperform the ones obtained by competitive systems, they do, however, surpass the results obtained by other summarization focused systems. Moreover, the integration of HanaNLG, in the case of cross-document timeline generation, improves the results obtained without affecting the performance of the whole system.

Therefore, it can be concluded that all objectives proposed in Chapter 1 have been successfully achieved through the research conducted in this thesis.

7.2 Work in Progress and Future Work

This thesis work represents a small portion of the research area in NLG that has mainly contributed to the surface realization task. However, there is still much work to be done. On the one hand, there are some issues that are currently being tackled and, on the other hand, some aspects that will be addressed in the future. The former will be detailed in Section 7.2.1 while the latter will be described in Section 7.2.2.

7.2.1 Work in Progress

With the aim of broadening the scenarios explored for testing HanaNLG, we are currently analyzing the performance of HanaNLG in another scenario. In particular this scenario is focused on the generation of children stories given some characters and actions. The text generated in this scenario will be useful for the creation of new children's stories based on the personal tastes of users. Therefore, with the long term objective of creating this type of children's stories, as a first step, we analyze HanaNLG in the context of story generation and we propose a method for automatically identifying characters in fictional narratives. These two issues are detailed below:

- **Studying the adaptation of HanaNLG for story generation.** In a first instance, the research was focused on the generation of a story from different kinds of elements (i.e., verbs, nouns and adjectives) that are considered relevant in the input document. We conducted a preliminary analysis of the performance of the approach in two different tasks: (i) regeneration of a story in the form of an abstractive summary; and (ii) generating a new story (recreation) taking into account the structure and vocabulary of an existing story.
  For the generation of the stories, the vocabulary is provided by an external macroplanning stage. This stage uses positional language models for the selection of the elements that will be used in the generation process. In the case of the surface realization stage, HanaNLG$_{FLMR}$ (see Section5.5 of Chapter 5) is used to generate the sentences.
  We used the collection of children's stories described in Section 5.5 of Chapter 5 as the corpora. From these tales, we selected 67 to perform the experimentation, which was composed of the tasks previously mentioned (i.e., regenerating the story and recreating the story) and the addition of a baseline where a story is generated without using the macroplanning stage. A preliminary evaluation of these stories was conducted not focusing on the coherence and structure of the text generated but the content. In this sense, we took as a quality indicator the variation of words in the documents

generated as well as in the original ones. This variation was computed as the ratio between the total number of words and the number of unique words elements (e.g., verbs or nouns) in a story. In addition to this, ROUGE metric was used in the case of the regeneration experiment to compare the content of the regenerated story to the original one. Table 7.1 summarizes the results obtained.

	Regeneration task			Recreation task		
	Baseline (no DP)	Macro-planning	Original	Baseline (no DP)	Macro-planning	Original
General Variation	54.34	34.61	61.06	59.49	34.43	61.06
Verb V.	40.80	18.15	61.74	64.23	26.09	61.74
Noun V.	55.49	40.02	55.49	55.94	36.96	55.49
Adjective V.	73.97	42.64	78.83	59.08	39.79	78.83
ROUGE-1 R.	47.00	52.00	–	–	–	–

Table 7.1: Results for the regeneration and recreation task (%) using HanaNLG$_{FLM}$. Higher values imply a richness variety of words, being the output in these cases better.

As mentioned before, the results shown in the table represent the variations of words within the stories generated. In the context of this evaluation we are analyzing the diversity of the language contained in the generated stories. The high values depicted in this table imply a rich variety of words within the corresponding grammatical category, that is, the higher the value the more diverse is the language of the generated text. Therefore, these preliminary results indicate that the generated stories, in terms of content, are promising, thus opening an interesting research line.

- **Analyzing the challenge of the computational identification of characters in fictional narratives.** Since the analysis of HanaNLG in the scenario of story generation yielded promising results, we wanted to investigate ways of automatically identifying characters in fictional narratives. In this sense, this type of research could be useful for analyzing which type of factors and situations are required in a narrative in order that a specific character appears. This analysis could lead to the generation of improved stories since it could be possible to more effectively adapt the generation of text associated to a character in each situation.

  In this context, the automatic identification of characters is addressed as a supervised binary classification task employing machine learning techniques. The use of this type of technique will help to analyze whether a potential noun within a story can be classified as a character or not.

  During the experimentation, several machine learning models were tested, the best one achieving an F-measure of around 0.83. These results mean that the tasks of automatic character identification could be integrated as part of a computational creativity processes or systems.

7.2.2 Future Research Directions

For the long term there some aspects that can be addressed that would benefit and improve HanaNLG. These aspects are as follows:

- **Analyzing other type of multi-lingual resources for the generation of text in different languages.** At this moment, the generation of text in HanaNLG is only possible for English, in the case of the complete version of the approach, due to the fact that some of the resources used in the development of our approach are language-dependent. Therefore, the search, the research and the analysis of other resources that would allow the generation of text in different languages is essential. These resources would need to contain both, linguistic and semantic information as in the case of VerbNet.

- **Researching and analyzing deep learning approaches for their inclusion in HanaNLG.** The use of deep learning approaches in the NLP field has increased in recent years. In the same way, some NLG approaches employ deep learning techniques that have recently emerged. However, in the case of NLG, these techniques have not been sufficiently tested. Therefore, we want to analyze whether their inclusion in our approach may improve the results obtained and the quality of the language generated. We would introduce this type of technique in the *Preprocessing* or the *Generation of sentences* modules, in order to help the creation of language models and also in the generation of language.

- **Researching and Analyzing the generation of longer and more complex sentences.** At present, HanaNLG generates sentences whose length is short, due to the structure of the frames used. Therefore, there is a research need to improve the generated language in terms of length or complexity. In this regard, it is essential for the search of resources to contain information (e.g., more detailed information about the components of the sentence) that enables an adaptation of our approach to generate longer sentences, with a complex sentence structure (e.g., including subordinate sentences).

7.3 Relevant Publications

Although some of these publications have been mentioned throughout this thesis, the following list groups them according to their related topic. To summarize, the total number of publications in journals is: 4 (2 of them are indexed in the Journal

Citation Reports, in the first and third quartile; and the remaining are indexed in Scopus) and the total number of publications in conferences and workshops is: 14 (including many prestigious conferences of the NLG and NLP area such as INLG, ENLG, CICLING, NLDB or SEPLN).

- Publications concerning the state-of-the-art (Chapters 2 and 3):
  - Marta Vicente, Cristina Barros, Fernando S. Peregrino, Francisco Agulló and Elena Lloret. 2015. La generación del lenguaje natural: análisis del estado actual. *Computación y Sistemas*. 19(4). pp 721-756.
- Publications regarding HanaNLG hybrid surface realization approach and its intrinsic evaluation (Chapters 4 and 5):
  - Cristina Barros and Elena Lloret. 2019. HanaNLG: A Flexible Hybrid Approach for Natural Language Generation. In *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing* (CICLING 2019). In press.
  - Cristina Barros and Elena Lloret. 2018. Surface Realisation Using Factored Language Models and Input Seed Features. In: *Castro F., Miranda-Jiménez S., González-Mendoza M. (eds) Advances in Computational Intelligence. MICAI 2017. Lecture Notes in Computer Science*, vol 10633. Springer, Cham
  - Cristina Barros. 2017. Estudio de un enfoque híbrido para la Generación del Lenguaje Natural. In *Proceedings of the Doctoral Symposium of the XXXIII International Conference of the Spanish Society for Natural Language Processing* (SEPLN 2017).
  - Cristina Barros, Dimitra Gkatzia, and Elena Lloret. 2017. Improving the Naturalness and Expressivity of Language Generation for Spanish. In *Proceedings of the 10th International Conference on Natural Language Generation* (INLG 2017). pp 41-50.
  - Cristina Barros, Dimitra Gkatzia, and Elena Lloret. 2017.Inflection Generation for Spanish Verbs using Supervised Learning. In *Proceedings of the 1st Workshop on Subword and Character Level Models in NLP* (SCLeM 2017). pp 136-141.
  - Cristina Barros and Elena Lloret. 2017. A multilingual multi-domain data-to-text natural language generation approach. *Procesamiento del Lenguaje Natural*. 58, pp 45-52.
  - Cristina Barros and Elena Lloret. 2017. Analysing the influence of semantic knowledge in natural language generation. In *Proceedings of the 12th International Conference on Digital Information Management* (ICDIM 2017). pp. 185-190.

  - Cristina Barros. 2016. Aproximación Híbrida para la Generación del Lenguaje Natural. In *Proceedings of the Doctoral Symposium of the XXXII International Conference of the Spanish Society for Natural Language Processing* (SEPLN 2016)
  - Cristina Barros and Elena Lloret. 2016. Generating sets of related sentences from input seed features. In *Proceedings of the 2nd International Workshop on Natural Language Generation from the Semantic Web* (WebNLG2016). pp. 1-4
  - Cristina Barros and Elena Lloret. 2015. Aproximación híbrida para la Generación del Lenguaje Natural. In *Primeras Jornadas de Investigadores Noveles.*
  - Cristina Barros and Elena Lloret. 2015. Input Seed Features for Guiding the Generation Process: A Statistical Approach for Spanish. In *Proceedings of the 15th European Workshop on Natural Language Generation* (ENLG2015). pp. 9-17.
  - Cristina Barros and Elena Lloret. 2015. Statistical NLG based on phonemes: a preliminary algorithm. In *Proceedings of the 1st International Workshop on Natural Language Generation from the Semantic Web* (WebNLG2015).
  - Cristina Barros and Elena Lloret. 2015. Proposal of a Data-to-text Natural Language Generation Approach to Create Stories for Dyslalic Children. In *Proceedings of the 1st Workshop on Data-to-text Generation.*

- Publications related to the extrinsic evaluation of HanaNLG and Work in Progress (Chapters 6 and 7):

  - Cristina Barros and Elena Lloret. 2019. HeadlineNLGen: A Hybrid Natural Language Generation Approach for Abstractive Headline Generation. *Information Sciences.* Submitted.
  - Cristina Barros, Marta Vicente and Elena Lloret. 2019. Tackling the Challenge of Computational Identification of Characters in Fictional Narratives. *IEEE International Conference on Cognitive Computing* (ICCC 2019). In Press.
  - Cristina Barros, Elena Lloret, Estela Saquete and Borja Navarro-Colorado. 2019. NATSUM: Narrative abstractive summarization through cross-document timeline generation. *Information Processing and Management.* 56 (5), 1775-1793 https://doi.org/10.1016/j.ipm.2019.02.010
  - Marta Vicente, Cristina Barros and Elena Lloret. 2018. Statistical language modelling for automatic story generation. *Journal of Intelligent and Fuzzy Systems*, 34(5), 3069-3079.

- Marta Vicente, Cristina Barros and Elena Lloret. 2017. A Study on Flexibility in Natural Language Generation Through a Statistical Approach to Story Generation. In *Proceedings of the 22nd International Conference on Applications of Natural Language to Information Systems* (NLDB 2017). pp. 492-498

APPENDIX A

Resumen

Este anexo contiene un resumen extendido en castellano de la investigación llevada a cabo en esta tesis doctoral que se enmarca en la tarea de GLN. En él se presenta primero los motivos subyacentes así como los objetivos perseguidos por los que se realizaron esta investigación. Seguidamente, se describen los métodos y sistemas más relevantes tanto para la GLN como para su evaluación. A continuación se presenta nuestra contribución al estado la cuestión a través de HanaNLG, el método propuesto para abordar la tarea de la GLN. Este método se evalúa intrínsecamente en diferentes dominios de manera incremental, donde los resultados obtenidos en un experimento justifican las decisiones tomadas en los siguientes. Asímismo, también se evalúa el método de manera extrínseca a través de su adaptación a otras tareas del Procesamiento del Lenguaje Natural (PLN). Finalmente, se destacan las conclusiones más importantes extraídas, los trabajos que se están desarrollando en la actualidad y se proponen las líneas de investigación futuras.

A.1 Introducción

A medida que la sociedad avanza, está surgiendo una nueva era de ecosistemas digitales, en la que se crean entornos colaborativos entre humanos y máquinas. Debido a esto, la comunicación y la interacción entre las personas y las máquinas debe ser lo más sólida, precisa y natural posible (Jacko, 2012). La tecnología se incluye en muchos aspectos de la vida cotidiana de los seres humanos, ya sea en los automóviles, los teléfonos móviles, los ordenadores o los televisores. Esta comunicación puede implicar diferentes niveles de dificultad dependiendo de cómo se realice. En este sentido, cuando la comunicación se establece en el dominio de la máquina (por ejemplo, utilizando un lenguaje de programación o pulsando un botón de la interfaz de una aplicación), la ambigüedad es imposible debido a las reglas implícitas en este tipo de comunicación. Sin embargo, cuando la comunicación es al revés, una máquina no puede manejar con facilidad la flexibilidad y ambigüedad del lenguaje natural (por ejemplo, cuando se pide en voz alta al teléfono móvil algún tipo de información o cuando se busca algo en internet a través de un motor de búsqueda).

Dentro del campo de la inteligencia artificial, el PLN se encarga del análisis automático y la representación del lenguaje humano, facilitando la comunicación entre personas y máquinas (Cole, 1997). Este campo de investigación proporciona las técnicas necesarias para comprender y generar lenguaje natural. En este contexto, se puede distinguir entre Comprensión del Lenguaje Natural (CLN) y GLN. El primero se refiere generalmente a la búsqueda, recuperación, clasificación y extracción de información, mientras que el segundo tiene por objetivo generar y emitir información apropiada de la manera más adecuada en base a un objetivo comunicativo previamente establecido (por ejemplo, informar, resumir, informar, persuadir, promover, alentar o ayudar).

El campo del PLN cubre una amplia gama de tareas, entre las que podemos destacar la traducción automática (Tantuğ & Adalı, 2018), los sistemas de recuperación de información (Berger & Lafferty, 2017), generación automática de resúmenes (Hardy & Vlachos, 2018) o GLN (Munigala et al., 2018). Cada una de ellas aborda el lenguaje natural de diferentes maneras, procesándolo automáticamente y teniendo en cuenta varios niveles de análisis del lenguaje, como por ejemplo: (i) análisis fonético y fonológico; (ii) análisis léxico morfológico; (iii) análisis sintáctico; (iv) análisis semántico; o (v) análisis pragmático.

Hasta ahora la mayor parte del esfuerzo de investigación en PLN se ha centrado en la CLN, relegando la tarea de la GLN a la mera extracción de fragmentos literales de texto (Vodolazova et al., 2013), técnicas de copiar y pegar (Jing & McKeown, 2000), el uso de plantillas (Mitchell et al., 2014) y enfoques específicos del dominio que generan lenguaje por medio de vocabularios y gramáticas restringidas (Bouayad-Agha et al., 2012; Androutsopoulos et al., 2013). Por lo tanto, y dado que el campo de la PLN ha abordado en mayor profundidad la CLN, en esta tesis, nos centramos en la tarea de GLN.

A.2 Motivación

Entre las disciplinas incluídas en el campo del PLN, la GLN es la encargada de desarrollar automáticamente técnicas para producir lenguaje humano, en forma de texto o habla (Bateman & Zoch, 2003). La investigación y el desarrollo de la tarea de la GLN incluye conocimientos de diferentes áreas como la lingüística, la psicología, la ingeniería y la informática. Como se ha mencionado anteriormente, el objetivo principal de esta disciplina es investigar cómo producir automáticamente textos de alta calidad en lenguaje natural. Para ello, puede partir de representaciones de datos estructuradas y que puedan ser procesadas (como las bases de datos) o de textos escritos en lenguaje natural.

Tradicionalmente y debido a la complejidad de esta tarea, los sistemas de GLN han utilizado enfoques híbridos, que combinan diferentes técnicas (Bangalore & Rambow, 2000). Sin embargo, una de las limitaciones de los sistemas actuales de GLN es que han sido diseñados para dominios(Ramos-Soto et al., 2015) y propósitos (Ge et al., 2015) muy específicos. Por lo que el desarrollo de enfoques de dominio abierto y flexibles supondrían un gran avance para el campo de la GLN (Barros & Lloret, 2017). Además, actualmente hay un nuevo reto añadido a esta tarea. Este reto está relacionado con el gran número de fuentes de información heterogéneas pertenecientes a diferentes géneros textuales (por ejemplo, noticias, blogs, reseñas, foros, redes sociales, etc.). Por lo tanto, es indispensable estudiar estas fuentes para comprender las características de cada una de ellas y poder diseñar métodos que sean independientes de las fuentes de datos, el dominio y el género textual al que pertenecen.

Partiendo de las limitaciones de los sistemas existentes, donde su diseño impide su adaptación a otros dominios y propósitos, la hipótesis de partida de esta tesis es que la aplicación de un enfoque híbrido para la generación de lenguaje natural aumentará la calidad del lenguaje producido, favoreciendo su independencia de dominio, de género textual y de la aplicación final que lo utiliza.

A.3 Objetivos

El objetivo principal de este proyecto de tesis doctoral es el análisis, investigación y propuesta de una aproximación híbrida para la GLN, que combine métodos estadísticos y basados en conocimiento. Para lograr este objetivo, se plantean una serie de objetivos más específicos, que permitirán la consecución del objetivo principal anteriormente expuesto:

1. Realizar un exhaustivo estado de la cuestión en el campo de la GLN, analizando los enfoques existentes en la actualidad.

2. Investigar, proponer y analizar nuevas aproximaciones para la GLN utilizando para ello técnicas basadas en PLN, centrándose principalmente en enfoques de GLN híbridos que combinen diferentes métodos estadísticos y basados en conocimiento.

3. Evaluar de manera exhaustiva el enfoque propuesto, utilizando métricas estándar o bien adaptando las metodologías de evaluación a las características de distintos dominios o escenarios. La evaluación realizada constará tanto de evaluaciones intrínsecas y extrínseca, así como también cuantitativas y cualitativas y se comparará en la medida de lo posible con las aproximaciones existentes.

4. Estudiar la aplicación del enfoque propuesto enmarcada en el contexto de otras tareas de PLN.

5. Conclusiones y beneficios de esta investigación junto con una propuesta de trabajos futuros.

A.4 Organización del trabajo

Este capítulo se ha organizado de la siguiente manera: la sección A.5 presenta brevemente el estado de la cuestión en el área de la GLN, resaltando la arquitectura habitual de un sistema de generación, los efoques más empleados para abordar la GLN y los métodos de evaluación para dicha tarea. Las secciones A.6 y A.7 hacen referencia a HanaNLG, nuestra propuesta de enfoque híbrido para la fase de realización. En la sección A.6 se presentan los módulos que conforman la estructura de HanaNLG y se detalla el proceso de generación junto con las técnicas y métodos involucrados. En la sección A.7 se expone la evaluación intrínseca incremental llevada a cabo junto con los experimentos realizados, donde los resultados de un experimento justifican las decisiones de los siguientes. Además de la evaluación intrínseca efectuada, en la misma sección (sección A.7) se muestra la adaptación de HanaNLG a otras tareas del PLN, concretamente a la tarea de generación automática de resúmenes. Este tipo de evaluación nos permitirá evaluar extrínsecamente el desempeño del enfoque propuesto. Finalmente, la sección A.8 contiene las conclusiones extraídas de este trabajo de tesis, así como los trabajos que se están llevando a cabo actualmente y el trabajo que se pretende realizar de cara al futuro.

A.5 Estado de la cuestión

Desde su concepción en los inicios de los años 50, la tarea de la GLN se ha extendido ampliamente. El objetivo principal de esta tarea, como se ha mencionado anteriormente, es producir de forma automática estructuras correctas de lenguaje natural a partir de una representación de la información (Cole, 1997).

Los sistemas de GLN se pueden clasificar en base a diversos factores. Por un lado, dependiendo del tipo de entrada al sistema, los sistemas de GLN pueden clasificarse en dos tipos: datos-a-texto y texto-a-texto. La diferencia entre estos dos tipos de sistema radica en que el primero (datos-a-texto), la entrada al sistema es un conjunto de datos que no forman un texto por ellos mismos (por ejemplo, datos numéricos provenientes de un sensor que representan los signos vitales de un paciente); mientras que en el segundo (texto-a-texto), el sistema toma como entrada un texto o documento textual de cuyo contenido (información relevante) se producirá un nuevo texto. Por otro lado, los sistemas de GLN se pueden clasificar según el objetivo del sistema (el propósito por el que el sistema fue creado). Por ejemplo, el objetivo del sistema no sería el mismo para un sistema que generase resúmenes que para un sistema que genere textos persuasivos. Existen diversos tipos de objetivos pero los más relevantes son: generación de textos informativos; generación automática de resúmenes; generación de textos simplificados; generación de textos persuasivos; sistemas de diálogos; generación de explicaciones de razonamiento; y generación de recomendaciones.

A.5.1 Arquitectura de un sistema GLN

Durante el desarrollo de un sistema de GLN es importante considerar para qué propósito se va a crear el sistema, qué vamos a tomar como entrada al sistema y qué arquitectura debería de tener. Con respecto a esto último, en una arquitectura se especifican las tareas o módulos necesarios para poder llevar a cabo el proceso de generación. Se han propuesto muchas arquitecturas distintas para abordar la tarea de la GLN (Kantrowitz & Bates, 1992; Hovy, 1988; Calder et al., 1999; García Ibáñez et al., 2004; Mellish et al., 2006), pero la que más se ha empleado hasta ahora es la propuesta por Ehud Reiter y Robert Dale (Reiter & Dale, 2000). De acuerdo a dicha arquitectura, las funcionalidades correspondientes a un sistema de GLN se distribuirían en 7 tareas distintas, agrupadas en una arquitectura básica de 3 módulos o fases, tal y como se muestra en la Figura A.1. Se considera que esta arquitectura es secuencial, por lo que los procesos de transformación de la información se realizan unidireccionalmente y secuencialmente. Esto da lugar a que no se puedan modificar los cambios efectuados en fases o tareas anteriores.

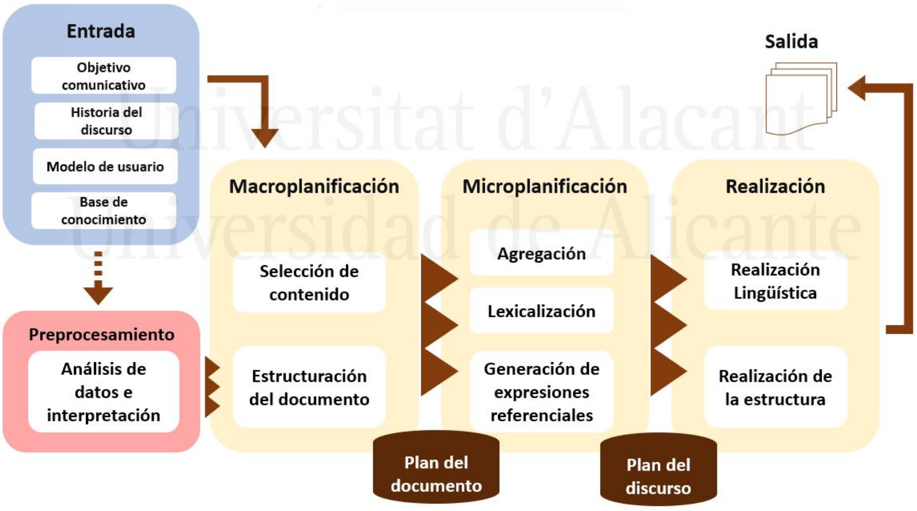


Figure A.1: Arquitectura de referencia de un sistema de GLN

A continuación se detallan cada una de estas fases junto con sus tareas correspondientes.

- **Macroplanificación**
  La fase de macroplanificación, también conocida como planificación del

  documento, comprende aquellas tareas en las que se decide qué información debería contener el texto final así como la estructura final que del texto, es decir, cómo se organizaría esta información. Como se mecionó en el apartado A.5, los sistemas de GLN pueden clasificarse atendiendo al tipo de entrada al sitema, pudiendo diferenciar entre los sistemas texto-a-texto y datos-a-texto. En el caso de estos últimos, algunos autores (Reiter, 2007) han añadido una fase previa de preprocesado para poder interpretar y analizar los datos de entrada al sistema antes de ejecutar la fase de macroplanificación. A la salida de esta fase se le denomina "*plan del documento*", el cuál suele tomar la forma de un árbol, en cuyos nodos finales se pueden encontrar los mensajes. Estos mensajes son unidades elementales del discurso del dominio para el cual se está generando texto que incluye información sobre las frases a generar así como la relación existente entre ellas.
  Las tareas llevadas a cabo en esta fase son las siguientes:

  - **Selección del contenido**
    La tarea de selección de contenido es la que permite al sistema elegir y obtener la información que debería ser comunicada en el texto final. Esta información sería la más relevante para el usuario de acuerdo al objetivo comunicativo y la situación. Esta situación incluye aspectos tan diversos como el tamaño de la salida del sistema, el nivel de conocimiento del usuario o el texto que se haya generado hasta el momento.

  - **Estructuración del documento**
    Esta tarea es la encargada de seleccionar la estructura que tendrá la salida final del sistema de GLN. Para poder estructurar correctamente un texto, la coherencia y la cohesión son claves. Con respecto a la coherencia, ésta permite concebir el texto como semánticamente correcto. En el caso de la cohesión, ésta es la propiedad que permite relacionar cada elemento del texto, ya sean palabras, frases o párrafos. A la hora de determinar la estructura de un texto, dicha estructura no sería la misma en el caso de un texto explicativo que la de una conversación o la de un texto comparativo donde varias propuestas son comparadas.

- **Microplanificación**
  La fase de microplanificación toma como entrada el "*plan de documento*" que se obtiene como salida de la fase de macroplanificación. El objetivo principal de esta fase es el de realizar diversas operaciones (que serán detalladas más abajo), principalmente lingüísticas, a los mensajes contenidos en el "*plan del documento*". Estas operaciones pueden tomar como recursos bases de conocicimento, ontologías u otros recursos lingüísticos; y tienen en consideración el objetivo comunicativo y la caracterización del

usuario (el modelo de usuario) para realizar adecuadamente sus elecciones (por ejemplo, si el texto va dirigido a un adulto o a un niÃśo; o en el caso de que se generen textos simplificados,la elección del vocabulario sería distinta). La salida de esta fase es el "*plan del discurso*", el cual contiene la información del "*plan de documento*" junto con los cambios realizados por las distintas tareas de esta fase.
En esta fase se realizan las siguientes tareas:

- **Agregación**
  En la tarea de agregación se determinan las estructuras del "*plan del documento*" que deben ser combinadas así como el orden de las mismas. Estas combinaciones pueden variar dependiendo de lo que el sistema necesite. Por ejemplo, algunos autores buscan eliminar la redundancia (Dalianis, 1996) y otros la combinación de varios mensajes en uno solo (Cheng et al., 1997). En cualquier caso, en esta tarea se intentará que lo generado sea conciso y sintácticamente simple al igual que coherente (Bernardos, 2007).
- **Lexicalización**
  La lexicalización en el ámbito de la GLN es la tarea responsable de seleccionar las palabras más adecuadas o las estructuras sintácticas concretas con las que referirse al contenido selecionado en fases previas. Exiten diversos tipos de variaciones para un mismo mensaje, como por ejemplo las variaciones de categoría sintáctica o variaciones semánticas (Reiter & Dale, 2000). Cuando esto ocurre y existen diversas opciones para un mensaje, se deben considerar aspectos como el nivel de formalidad (*padre* en vez de *papá*) o las preferencias del usuario.
- **Generación de expresiones referenciales**
  La función principal de la tarea de generación de expresiones referenciales es determinar de qué manera van a ser referenciadas las entidades o conceptos que forman parte del "*plan del documento*" para evitar la redundancia. Por lo que, según Reiter y Dale (Reiter & Dale, 2000), estas expresiones referenciales deberían incluir la información necesaria que permita identificar unequívocamente a un elemento del discurso (ya sea una entidad, un concepto, etc.).

- **Realización**
  Dentro de la arquitectura propuesta en (Reiter & Dale, 2000), esta fase corresponde a la última, tomando como entrada el "*plan del discurso*" obtenido de la fase de microplanificación. El principal objetivo de la realización es el generar la salida final del sistema dotándola de una estructura correcta y un formato específico a las oraciones la componen. Por lo tanto, en esta fase se abordan aspectos como la sintaxis, la morfología o la ortografía. En esta fase se distinguen dos tipos de tareas:

  - **Realización lingüística**
    La realización lingüística es la tarea encargada de transformar los mensajes contenidos dentro del "*plan del discurso*" a texto en lenguaje natural.
  - **Realización de la estructura**
    El último paso dentro del proceso de la GLN estaría condicionado por la aplicación final del sistema. Por lo que, el objetivo de esta tarea sería adecuar la salida del sistema a un formato concreto. Por ejemplo, si el texto generado fuese a mostrarse dentro de una página web, éste podría requerir de algunas etiquetas HTML.

A.5.2 Enfoques para abordar la tarea de la GLN

A continuación se describen os enfoques más comunes para abordar la tarea de la GLN.

- **Enfoques basados en conocimiento**
  El conocimiento, dentro de los enfoques basados en conocimiento, suele estar representado explícitamente mediante el uso de herramientas como las ontologías, conjuntos de reglas o teasauros (diccionarios de sinónimos). Los sistemas que utilizan este tipo de enfoques suelen estar compuestos de dos subsistemas distintos: (i) una base de conocimiento y (ii) un motor de inferencia. Una base de conocimiento es un tipo de base de datos de gestión del conocimiento que proporciona los medios necesarios para la recopilación, organización y recuperación de conocimiento. En cambio, un motor de inferencia es la parte del sistema que, dada una secuencia, razona empleando el contenido de la base de conocimiento. Este motor examina cada una de las reglas de la base de conocimiento y realiza la acción pertinente cuando la condición de alguna de esas reglas se cumple. En este tipo de enfoques, la sistematización del conocimiento suele basarse en teorías lingüísticas, siendo las más comunes las siguientes: (i) la teoría de la estructuración retórica (Mann & Thompson, 1988); (ii) la gramática sistémico funcional (Halliday, 1985); (iii) gramática de adjunción de árboles (Joshi & Schabes, 1997); (iv) Teoría del sentido-texto de Mel'čuk (Mel'cuk et al., 1988); y (v) la teoría del centrado (Grosz & Sidner, 1986; Grosz et al., 1995).
  Este tipo de técnicas se han empleado a través de todo el proceso de GLN. Por ejemplo, la macroplanificaión es abordada, en el sistema presentado en (McDonald, 2010), utilizando operadores retóricos, para transformar los objetivos comunicativos en un árbol donde los nodos terminales son las proposiciones y los operadores son las reglas de derivación del árbol. Con respecto a la microplanificación, la tarea de agregación ha sido tratada utilizando un conjunto de reglas y unidades de información que proporcionan una salida única (Dalianis, 1996) o se ha basado en la exploración de

árboles de dependencia y en la teoría de la estructuración retórica (Theune et al., 2006). La fase de realización ha sido interpretada desde la teoría del sentido-texto como un paso final en una secuencia de transformaciones llevadas a cabo en representaciones lingüísticas, las cuales pueden ser tratadas a través de gramáticas o reglas que permitan la traducción de grafos (Wanner et al., 2010). Otro ejemplo de sistemas que emplean este tipo de técnicas, en más de una fase, es el presentado en (Gong et al., 2017). Este sistema genera un informe de prensa con el uso de reglas y plantillas utilizando la herramienta descrita en (Zock & Lapalme, 2010).

- **Enfoques estadísticos**
  Los enfoques estadísticos están basados en las probabilidades extraídas de un volumen de texto base, ya sea un corpus —anotado o no—, texto de la web, etc. Una de las herramientas primarias de este tipo de enfoques son los modelos de lenguaje. Un modelo de lenguaje estadístico es un mecanismo que define la estructura del lenguaje, o lo que es lo mismo, que toma como válidas secuencias de palabras en base a su frecuencia de aparición en un conjunto de textos. Esta frecuencia de aparición suele expresarse empleando distribuciones de probabilidad. Un buen modelo de lenguaje sería capaz de aceptar frases en base a su probabilidad (si están bien construidas y su probabilidad es alta) o rechazarlas en el caso de que su probabilidad sea baja. En el ámbito de la GLN, los modelos de lenguaje más utilizados son los siguientes: (i) modelo de n-gramas; (ii) modelos basados en gramáticas estocásticas; y (iii) MLF (estos modelos serán desarrollados en la sección A.6 ya que son empleados dentro del enfoque de GLN propouesto).
  Este tipo de enfoques no suelen emplearse en todas las fases que constituye la arquitectura de un sistema de GLN, pero sí que se usan en las dos últimas fases. Por ejemplo, el sistema presentado en (Ballesteros et al., 2014) aborda la tarea de lexicalización desarrollando un generador estadístico capaz de seleccionar los términos corrpespondientes a un conjunto de representaciones semánticas con el uso de clasificadores y el AnCora-UPF treebank (Mille et al., 2013). Con respecto a la tarea de generación de expresiones referenciales, el sistema mCRISP (Garoufi & Koller, 2011) genera expresiones referenciales gracias al uso de clasificadores que han sido entrenados sobre un corpus de descripciones. En el caso de la fase de realización, uno de los primeros trabajos que presentaban este tipo de téscnicas fue el desarrollado por *Langkilde y Knight* en 1998 (Langkilde & Knight, 1998). En este sistema, los modelos n-gramas son usados para determinar las variaciones de las palabras (ya sea el usar el plural o no, género, etc.), y las que tienen la mayor probabilidad dentro del modelo son usadas en la salida final del sistema. En (Barros & Lloret, 2017), la fase de realización es llevada a cabo utilizando MLF para generar frases en distintos dominios.

- **Enfoques híbridos**
  Se les suele denominar enfoques híbridos a aquellos enfoques que combinan las técnicas empleadas en los enfoques basados en conocimiento con las utilizadas en los enfoques estadísticos. Desde finales del siglo XX podemos encontrar ejemplos de este tipo de sistemas. El sistema FERGUS (Bangalore & Rambow, 2000) fue uno de los primeros sistemas híbridos desarrollados para GLN, el cual realiza sólo dos de las fases de la arquitectura descrita, las fases de microplanificiación y la de realización. Este sistema genera texto combinando modelos de n-gramas con un modelo estadístico basado en árboles y una gramática sintáctica lexicalizada, que se basa en gramáticas XTAG. La aplicación FLIGHTS (White et al., 2010) es otro ejemplo de sistema híbrido que presenta la información sobre vuelos de una forma personalizada para cada usuario (por ejemplo, considerando si un usuario es un estudiante o un viajero frecuente). Para generar texto utiliza diferentes bases de conocimiento (modelos de usuarios, modelos de dominio o registros de diálogos) para seleccionar el contenido del texto final y la herramienta OpenCCG[1] (que emplea modelos de lenguaje n-gramas y MLF internamente) para generar el texto final.
  Más recientemente, (Mille et al., 2016) presenta una propuesta preliminar de sistema multilingüe para la generación de resúmenes abstractivos que utiliza representaciones semánticas. Este sistema, cuyo marco teórico subyacente es la teoría del sentido texto, combina técnicas estadísticas y basadas en reglas para producir un resumen en respuesta a una consulta de usuario. Gardent y Perez-Beltrachini (Gardent & Perez-Beltrachini, 2017) proponen un enfoque híbrido simbólico/estadístico para modelar las limitaciones que regulan las interacciones de grano fino exixtentes entre las tareas de un sistema de GLN. Este enfoque utiliza una gramática genérica escrita a mano, un *hypertagger* estadístico y un algoritmo de realización para alcanzar este propósito. Con respecto al investigación realizada para el español, en (García-Méndez et al., 2018) se propone un sistema híbrido para generar frases a partir de pictogramas. Este sistema combina conocimiento lingüístico dado por un lexicon y un modelo de lenguaje para inferir proposiciones. Entonces, este conocimiento, en conjunto con una adaptación al español de SimpleNLG (Gatt & Reiter, 2009), es usado para generar frases coherentes.

- **Enfoques basados en aprendizaje profundo**
  En los últimos años, el apredizaje profundo ha ganado popularidad en todo el campo del PLN. Asimismo, en el ámbito de la GLN han surgido algunos trabajos en los últimos dos anõs. Hasta donde sabemos, este tipo de enfoques no están lo suficientemente extendidos en el campo de la GLN. A pesar de que el número de trabajos existentes que usan este tipo de tecnología es más bajo que el número de trabajos que emplean los enfoques

[1] http://openccg.sourceforge.net/

clásicos (por ejemplo, los enfoques basados en conocimiento o estadísticos), cada vez son más los enfoques que están integrando o probando técnicas de aprendizaje profundo.

En este contexto, un ejemplo de este tipo de sistemas es el presentado en (Lebret et al., 2016), donde se propone un modelo neuronal para datos-a-texto. Este modelo, que está construido sobre modelos de lenguaje neuronales condicionales, genera frases biográficas a partir de biografías de Wikipedia. En (Brad & Rebedea, 2017) se presenta un enfoque neuronal para paráfrasis. Este enfoque usa modelos secuencia-a-secuencia con atención, en conjunto con *transfer learning*, y usa *textual entailment* y pares de paráfrasis de frases para la generación de paráfrasis. Recientemente, Castro Ferreira et al. (2018) presentó un enfoque para la generación de expresiones referenciales que hace decisiones sobre la forma y el contenido del texto generado sin hacer una extracción de características explícitamente.

Respecto a los sistemas de aprendizaje profundo para la tarea de GLN, sus salidas pueden incluir contenido incorrecto o añadir contenido que no está explícitamente en la entrada del sistema[2]. Esto puede no ser adecuado en algunas aplicaciones de la GLN, como por ejemplo en la generación de informes médicos o financieros, donde la información que se desea generar tiene que ser fiable y precisa. También existen otros casos donde el lenguaje generado no es correcto desde un punto de vista gramatical y el contenido no tiene sentido (Subramanian et al., 2017).

A.5.3 Evaluación de sistemas de GLN

Respecto a la evaluación de la GLN, existe un consenso general entre los investigadores de este campo sobre la dificultad que entraña debido a sus peculiaridades (Viethen & Dale, 2007). A diferencia de otras tareas del PLN, en el área de la GLN no están adecuadamente especificadas tanto la entrada al sistema como su salida. Además de esto no existe una única salida correcta, por lo que no hay definido un criterio para evaluar la calidad de la salida.

Cuando se evalúa un sistema de GLN se pueden seguir diferentes estrategias (Resnik & Lin, 2010). Podemos distinguir entre evaluación intrínseca y evaluación extrínseca. En el caso de la primera, se suele evaluar el desempeño y la eficiecia del sistema en sí. En el caso de las evaluaciones extrínsecas, se evalúa el impacto que tiene el sistema sobre los usuarios en otras tareas. También puede hacerse una diferenciación entre evaluación automática y evaluación manual.

En el caso de la evaluación automática, esta evaluación puede realizarse de en términos cuantitativos usando métricas para comparar el texto generado por un sistema de GLN con un texto ideal creado por un experto o con un corpus de referencia. Las métricas que se utilizan en estos casos suelen provenir de otras

[2] https://ehudreiter.com/2018/11/12/hallucination-in-neural-nlg/

áres del PLN y han sido adoptadas debido a sus buenos resultados en sus respectivos campos. Este tipo de evaluación automática basada en corpus es atractiva para la GLN, como en otras áreas de PLN, por su velocidad, reproducibilidad y su bajo coste computacional (Reiter & Belz, 2009). Algunas métricas empleadas en la GLN vienen del campo de la traducción automática. Este es el caso de las métricas BLEU (Papineni et al., 2002), NIST (Doddington, 2002) o METEOR (Lavie & Agarwal, 2007). Otras, como ROUGE (Lin, 2004), provienen del campo de generación automática de resúmenes. El uso de este tipo de métricas ha sido discutido en la comunidad de la GLN. Esto es debido a diversos factores entre los que se incluye que, a diferencia de otras áreas del PLN, en la GLN puede haber, para un mismo sistema, más de una salida válida o la poca cantidad de corpus específicos para evaluar esta tarea así como que las métricas son de otras áreas del PLN, por lo que pueden no ajustarse debidamente a la tarea de GLN(Scott & Moore, 2007). Además, está el hecho de que los resultados obtenidos por este tipo de métricas pueden ser difíciles de interpretar (Paris et al., 2007).

El uso de las métricas mencionadas puede no ser suficiente para evaluar algunos aspectos de un texto generado por un sistema de GLN, como su significado o corección. Por lo que, una evaluación manual podría ser más adecuada en estos casos. En este contexto, las evaluaciones basadas en calificaciones humanas o en preferencias son actualmente las más utilizadas en el campo de la GLN (Reiter & Belz, 2009). En este tipo de evaluación, se suele pedir a evaluadores humanos que califiquen los textos a través de encuestas, cuestionarios o plataformas de *crowdsourcing*. Este tipo de encuestas suelen estar compuestas de varias preguntas que dependen del aspecto del texto a ser evaluado. Las preguntas, dependiendo del tipo de evauación manual realizada, pueden diferir de una evaluación a otra. Por un lado, en las evaluaciones que usan calificaciones, suele evaluarse la calidad lingüística (es decir, la legibilidad y fluidez) y la calidad del contenido (es decir, precisión, adecuación, relevancia o corrección) utilizando escalas de calificación (por ejemplo, una escala de Likert de 5 puntos). Por otro lado, en las evaluaciones basadas en preferencias humanas, suele pedirse a los evaluadores que ordenen varios textos en base a sus preferencias (Belz & Kow, 2010). La principal preocupación de este tipo de evaluaciones reside en que la subjetividad de los evaluadores podría afectar la fiabilidad entre evaluadores, dando lugar a auqe haya una alta variabilidad en las evaluaciones realizadas por varios evaluadores (Gatt & Krahmer, 2018).

Además de los tipos de evaluación comentadas (evaluación intrínseca, extrínseca, automática y manual), los sistemas de GLN también pueden ser evaluados mendiante competiciones. En estas competiciones se suele realizar una evaluación colaborativa en la que se intenta resolver un problema específico o tarea concreta (en el caso de la GLN podrá ser una de las tareas de la arquitectura general descrita como la realización lingüística o la generación de expresiones refereanciales) cuya resolución deberían afrontar varios equipos de trabajo para finalmente comparar los resultados obtenidos (Viethen & Dale, 2007). Sin embargo, el uso de este tipo de evaluación colaborativa también ha sido discutida por la comunidad científica (Dale & White, 2007) y debido a la naturaleza compleja de esta disciplina algunos asuntos deberían tenerse en consideración. Por ejemplo, el tipo de tarea a evaluar, el tipo de métricas a usar o las bases metodológicas necesarias (tanto en términos de la competición en sí como en la comparación de los resultados).

A.6 HanaNLG: enfoque híbrido para la generación de lenguaje natural

HanaNLG es la principal contribución de esta tesis doctoral al campor de la GLN. Este enfoque es un enfoque híbrido para la fase realización capaz de generar automáticamente texto que es fácilmente adaptable a diferentes géneros, dominios y lenguajes. HanaNLG es híbrido porque se basa en el uso de recursos lingüísticos así como en información estadística, a través del uso de MLF, para construir la salida final. Para generar lenguaje, el enfoque propuesto hace uso de estrategias de *over-generation y ranking*, donde primero se genera un conjunto de frases candidatas para después realizar un ranking donde se seleccione una frase en base a un criterio definido, en nuestro caso, su probabilidad. Además, dado que HanaNLG solo está enfocado en la fase de realización, no tenemos información de los procesos de macroplanificación y microplanificación, para poder guiar la generación en base a un tema concreto, palabras, dominio, etc., proponemos el concepto de característica semilla. Estas características semillas pueden considerarse objetos abstractos (por ejemplo, fonemas, sentimientos, polaridades, etc.) que guiarán el proceso de generación en relación al vocabulario que la frase generada deba contener o estar relacionada. Por consiguiente, el tipo de textos generados por HanaNLG puede ser adaptado a diferentes dominios y también a diferentes objetivos comunicativos (por ejemplo, generación automática de resúmenes). Asímismo, dada la naturaleza de los recursos y técnicas empleadas, nuestro enfoque también es fácilmente adaptable a diferentes géneros, dominios y lenguajes.

La arquitectura de HanaNLG está compuesta por una arquitectura de 6 módulos, tal y como se ve en la figura A.2. Las entradas para el enfoque son: (i) corpus; (ii) una cacterística semilla; (iii) el número de frases a generar; (iv) el nivel de abstracción (es decir, utilizando palabras, lemas o *synsets*[3]); y, opcionalmente, (v) una variable *booleana* indicando la generación de frases relacionadas (es decir, frases cuyo sujeto u objeto están relacionados) y/o (vi) tiempos verbales de la frase (es decir, estos tiempos verbales serán los empleados durante el módulo de *Flexión de frase* si se ha proporcionado). Aunque los módulos que constituyen la arquitectura de este enfoque no siguen estrictamente las tareas definidas para la fase de realización, éstos fueron desarrollados de esta manera para asegurar que el proceso generación fuese lo más flexible y adaptable posible. Sin embargo,

[3] Conjunto de sinónimos cognitivos relacionados a un término o concepto usado en WordNet.

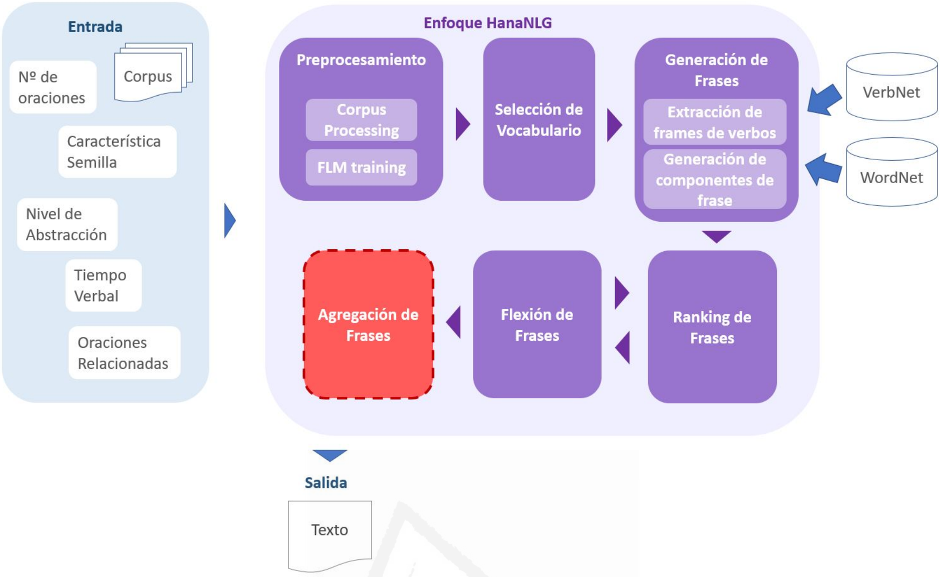


Figure A.2: Arquitectura de HanaNLG.

debido a que alguno de los recusos utilizados en el desarrllo de HanaNLG están sólo disponibles en inglés, el enfoque actual de HanaNLG sólo genera texto en inglés. En el caso de que esos recursos se creasen en otros idiomas, HanaNLG será capaz de generar texto en esos idiomas. Para la generación de cada frase se sigue el mismo proceso que se describirá a continuación.

A.6.1 Preprocesamiento

El primer paso antes de empezar el proceso de generación en HanaNLG es analizar los datos de entrada así como adaptarlos para poder ser usados. Dado que la mayor fuente de información proporcionada como entrada es un corpus, este será el recurso a analizar dentro de este módulo.

El corpus de entrada consiste en un conjunto de documentos en texto plano (por lo que no está etiquetado de ninguna manera), por lo que, en una primera instancia, se realiza un análisis lingüístico para poder obtener distinto tipo de información. Específicamente, utilizando la herramienta Freeling (Padró & Stanilovsky, 2012) se recoge información sobre las palabras en sí, su lema, su etiqueta gramatical (POS tag) e información semántica (synset). Esta información se usará para etiquetar automáticamente el corpus. Además, también se utilizará durante el módulo de Generación de frases así como para entrenar los MLF usados durante todo el proceso de generación.

Una vez que el corpus está etiquetado, se entrenan los modelos MLF sobre el corpus de entrada. En este tipo de modelos, una palabra es vista como un vector de k factores, tal que $w_t \equiv \{f_t^1, f_t^2, \ldots, f_t^K\}$, donde estos factores pueden ser cualquier cosa (como un POS tag, un lema, etc.). El objetivo principal de un MLF, en base a los factores seleccionados por el usuario, es crear un modelo estadístico $P(f|f_1,\ldots,f_N)$ donde la predicción de un factor f depende de sus N padres $\{f_1,\ldots,f_N\}$.

A.6.2 Selección del vocabulario

Otra de las tareas que se realizan antes de comenzar el proceso de generación es la selección del vocabulario que aparecerá en el texto final. El objetivo de este módulo es similar al que tendría la tarea de selección de contenido pero se diferencia de ella en que el vocabulario se obtendrá en base a la característica semilla dada como entrada. El vocabulario obtenido será almacenado en una bolsa de palabras para su posterior utilización.

HanaNLG puede manejar diferentes tipos de característica semilla teniendo, cada uno de ellas, rasgos distintivos. Por ejemplo, si se tomase como cacterística semilla un fonema, éste podría ser útil en la generación de frases para terapias del lenguaje donde se trabajan aspectos fonéticos y fonológicos.

La detección de este vocabulario en el corpus de entrada es una tarea compleja ya que depende del tipo de característica semilla. Algunas de ellas podrían requerir el uso de rescursos lingüísticos, como lexicones o corpus, o herramientas específicas que faciliten la detección de palabras relacionadas con la característica semilla (por ejemplo, en el caso de detectar palabras clave, se podrían utilizar sistemas de detección de tópicos o palabras clave).

A.6.3 Generación de frases

El objetivo principal de este módulo es la generación de un conjunto de frases candidatas para que en el siguiente módulo se realice un ranking con ellas. Estas frases pueden generarse empleando únicamente palabras o elementos más abstractos como los lemas o los synsets. La especificación de con qué elemento va a ser generada la frase viene definidio en el *nivel de abstracción* dado en la entrada al enfoque.

El proceso de generación en este enfoque se realiza desde el núcleo de la frase, es decir, en nuestro enfoque consideramos que el verbo es el núcleo central de una oración. Por lo tanto, en una primera instancia, haciendo uso de los recursos VerbNet (Schuler, 2005) y WordNet (Fellbaum, 1998), para un verbo dado (el cual puede estar dentro del vocabulario o ser de los verbos más frecuentes dentro del corpus de entrada) se extraen los *frames* sintácticos de cada una de estos recursos. Estos *frames* contienen información tanto semántica como sintáctica de una gran variedad de verbos. Además, los *frames* proporcionan la estructura que tendrá la frase a generar, indicando qué tipo de elementos (por ejemplo, sintagmas nominales, sintagmas adjetivales, etc.) necesita el verbo, evitando así el tener que definir o usar gramáticas, las cuales son muy costosas de crear.

Una vez se han extraído los *frames* del verbo, para cada uno de ellos se genera una frase atendiendo a los elementos necesarios que especifique el *frame*. Por ejemplo, el *frame* puede indicar que cierto verbo necesita un sujeto y que dicho sujeto debe ser un animal. En ese caso, haciendo uso de los MLF entrenados anteriormente, se buscaría un elemento que contenga esas cualidades y que tenga la mayor probabilidad de aparecer con el núcleo verbal, siempre priorizando las palabras contenidas en el vocabulario.

Este módulo también puede generar frases relacionadas con oraciones generadas con anterioridad. En estos casos, se analiza el *frame* para verificar si alguno de sus elementos (en relación con el tipo de sujeto, objeto, etc.) coincide con cualquiera de los elementos generados en la frase anterior. En el caso de que esto sucediera, el elemento coincidente pasaría a formar parte de la frase que se esté generando en ese momento.

A.6.4 Ranking de frases

El objetivo principal de este módulo es el de realizar un ranking con las frases generadas en el módulo anterior, para finalmente selecionar sólo una frase. La frase seleccionada pasará a formar parte de la salida final del sistema. Para ello, las frases son clasificadas en base a su probabilad. Esta probabilidad se calcula siguiendo la regla de la cadena, tal y como se muestra en la ecuación A.1.

$$P(w_1, w_2 ... w_n) = \prod_{i=1}^{n} P(w_i | w_1, w_2 ... w_{i-1}) \tag{A.1}$$

Basándonos en la regla de la cadena, la probabilidad de una frase puede ser calculada como el producto de las probabilidades de sus palabras. Dependiendo del modelo de lenguaje empleado, la probabilidad de una palabra puede calcularse de diferentes maneras. En nuestro caso, la probabilidad de una palabra es calculada como la combinación lineal de MLF, como se sugiere en (Isard et al., 2006). En esta combinación lineal, a cada MLF utilizado se le asigna un peso λ_i, siendo la suma total de todos los pesos 1. En la ecuación A.2 se muestra esta combinación lineal, donde f se refiere al factor usado para entrenar los MLF.

$$P(f_i | f_{i-2}^{i-1}) = \lambda_1 P_1(f_i | f_{i-2}^{i-1})^{1/n} + \cdots + \lambda_n P_n(f_i | f_{i-2}^{i-1})^{1/n} \tag{A.2}$$

A.6.5 Flexión de frases

Cuando hablamos de lenguaje natural, la flexión morfológica es clave ya que sin ella la información no puede ser correctamente transmitida y puede perderse la referencia del tiempo y persona que está realizando el acto de comunicación. Por este motivo, este paso es indispensable para poder hacer el lenguaje más natural y fluido. Por lo tanto, el objetivo de este módulo es flexionar la frase escogida en el anterior módulo antes de incluirla en la salida final del enfoque.

La flexión morfológica es una característica común a muchos idiomas que permite la concordancia en género y número. Las reglas para flexionar palabras puede variar de un lenguaje a otro. Por lo tanto, este módulo necesita ser entrenado o adaptado dependiendo del idioma objetivo. Independientemente del idioma utilizado durante la generación este módulo realizará cambios menores genéricos a las frases generadas dependiendo del *nivel de abstracción* (es decir, palabra, lema o synset) provisto como entrada.

En el caso de haber empleado palabras para generar la frase, los cambios a las palabras sólo afectaría a la concordancia en número (singular y plural) y a la persona (primera persona del singular, etc.). Estos cambios se realizarán en base a las características del verbo. Por otro lado, si la frase ha sido generada con lemas o synsets, hay algunas cuestiones que decidir. Con respecto a los lemas, el módulo ya tiene la palabra a flexionar, pero en el caso de los synsets sólo se tiene un identificador del conjunto de sinónimos en WordNet. Por lo tanto, es esencial decidir qué componente del conjunto se ha de escoger para poder flexionarlo. Para poder escogerlo, primero expandimos el synset en todos sus componentes (los sinónimos dentro del conjunto suelen estar en forma de lema). Después, se genera un conjunto de frases con todas sus palabras en lema con todas las combinaciones posibles usando lo sinónimos. Una vez tenemos el conjunto de frases con sus palabras en forma de lema, el proceso de flexión seguido es el mismo tanto para el caso de los lemas como para el de los synsets. Este proceso empieza con la flexión del verbo, la cual puede tomarse de los *tiempos verbales* introducidos en la entrada al enfoque o el módulo elegirá automáticamente el tiempo verbal probando todas las combinaciones de tiempo verbal y escogiendo la que haga que la frase tenga la mayor probabilidad posible. Una vez que el tiempo verbal está definido, se realizan los mismos cambios menores genéricos que en el caso de haber generado la frase con palabras.

Una vez que la frase se ha flexionado, pasaría a tomar parte de la salida final del enfoque.

A.6.6 Agregación de frases

Este último módulo del enfoque es opcional y sólo se ejecutaría si se especifica en la entrada y si se ha generado más de una frase. La agregación de frases puede ser necesaria al final del proceso de generación para evitar la repetición así como la redundancia de información en la salida final (Dalianis, 1999).

En este contexto, es posible que un mismo sujeto aparezca varias veces en frases consecutivas debido a la generación de frases relacionadas. Por lo tanto, la información contenida en estas frases podría combinarse en una sola frase. Para este propósito, realizamos una agregación basada en reglas. Concretamente, en este momento, la agregación sólo afectaría al sujeto y al verbo de las frases. Se han definido dos tipos de reglas:

- **Regla 1**: Dos frases consecutivas son combinadas cuando sus sujetos y verbos coincidan. Ejemplo:
  "María es joven. María es simpática."

“María es joven y simpática."

- **Regla 2**: Dos frases consecutivas son combinadas si sus sujetos coinciden pero sus verbos no. Ejemplo:
“María es joven. María viaja mucho."
"María es joven y viaja mucho."

Estas reglas son bastante similares pero sirven para el propósito de evitar la redundancia en la salida final de HanaNLG.

A.7 Evaluación de HanaNLG

Esta sección describe la evaluación llevada a cabo para comprobar el desempeño de HanaNLG. En este contexto, primero se realizó una evaluación intrínseca donde se evaluó el texto generado por HanaNLG en diferentes dominios. A consecuencia de los resultados obtenidos en esta evaluación intrínseca, se realizó una evaluación extrínseca probando HanaNLG en otros campos del PLN, concretamente en la tarea de generación automática de resúmenes.

A.7.1 Evaluación intrínseca de HanaNLG

Durante el diseño y desarrollo de los diferentes módulos de HanaNLG se plantearon una serie de cuestiones de investigación que se tuvieron en cuenta en la construcción final del enfoque. Estas cuestiones serán analizadas y tratadas en las siguientes preguntas de investigación. Por lo tanto, realizaremos una evaluación incremental donde las respuestas de una pregunta justificarán las decisiones tomadas en las preguntas subsiguientes.

A continuación vamos a mostrar para cada pregunta de investigación, los resultados obtenidos junto a una discusión de los mismos. Dado que la tarea de evaluar un texto generado por un sistema de GLN es compleja y dificil de realizar de manera automática, en la gran mayoría de los casos se hizo un evaluación manual de las frases generadas.

- **¿El uso de las características semilla permite la generación de lenguaje que pueda cumplir con un propósito u objetivo específico?**

  Las características semilla proporcionan flexibilidad con respecto al contenido del texto generado y, por lo tanto, es uno de los aspectos principales que sustentan el enfoque. Por este motivo, en una primera instancia, evaluamos la idoneidad de las características semilla para generar un texto que cumpla con los requisitos de un propósito u objetivo específico. Para ello, se ha construido un enfoque preliminar de generación para la fase de realización. Este enfoque, HanaNLG$_{N-gram}$, combina el uso de características semilla con modelos de lenguaje n-gramas para generar lenguaje.

Frases	**Porcentaje local (basado en 95 frases)**	**Porcentaje global (basado en 208 frases)**
Frases generadas con bigramas con (</s>)	46,32%	21,15%
Frases generadas con trigramas con (</s>)	78,95%	36,06%
Frases nuevas no incluidas en el corpus	73,68%	33,65%
Total de frases con sentido	56,84%	25,96%
Frases con sentido incluidas en el corpus	25,26%	11,54%
Frases nuevas con sentido no incluidas en el corpus	31,58%	14,42%
Frases nuevas con sentido generadas con bigramas	9,47%	4,33%
Frases nuevas con sentido generadas con trigramas	22,11%	10,10%

Table A.2: Estadísticas de las frases generadas por HanaNLG$_{N-gram}$ que terminan con (</s>).

En la Tabla A.2 se muestran los resultados de las frases generadas por HanaNLG$_{N-gram}$, utilizando una colección de cuentos de Hans Christian Andersen en inglés como corpus y fonemas como característica semilla. Estas estadísticas fueron calculadas en base al número total de oraciones diferentes que terminan con el símbolo "'fin de la oración" (</s>).

Como se muestra en la tabla, los resultados de las frases con sentido fueron prometedores. Casi la mitad de las frases que terminaron con el *token* (</s>) tienen sentido. Además, las frases con sentido representan alrededor del 30% de las 95 frases diferentes que no aparecen explícitamente en el corpus de entrada, es decir, se trata de frases de nueva creación. Además, hemos comprobado que cada frase generada contenía al menos una palabra relacionada con la característica semilla. Estos resultados son bastante positivos y muestran que el uso de características semilla dentro de un proceso simple de generación pueden conducir a la generación de frases que cumplan con un propósito u objetivo concreto.

- **¿Cuál es el modelo de lenguaje más adecuado para generar lenguaje en la arquitectura de HanaNLG?**

  Dado que la primera pregunta de investigación obtuvo resultados prometedores con el uso de n-grams como modelo de lenguaje, el objetivo de esta segunda pregunta de investigación es analizar otros modelos de lenguaje para el proceso de generación.

  Para ello, analizaremos el rendimiento de los MLF en contraste con el uso de n-grams en el proceso de generación. Por lo tanto, compararemos los resultados obtenidos de este análisis con los de HanaNLG$_{N-gram}$. Para esta experimentación se construyó un modelo de generación de la fase de realización, de manera similar a HanaNLG$_{N-gram}$, pero empleando MLF, al cual denominaremos HanaNLG$_{FLM}$.

Para esta investigación se han generado frases en inglés y en español para el escenario de tecnologías de asistencia, empleando los fonemas como característica semilla y usando una colección de cuentos de Hans Christian Andersen como corpus. En el caso de los MLF se han usado dos combinaciones distintas de factores en su entrenamiento: (i) Palabra + Etiqueta gramatical (*WP*) y (ii) Lema + Etiqueta gramatical (*LP*). La tabla A.3 resume los resultados obtenidos de la evaluación manual de las frases realizada.

Modelo de lenguaje		**Frases totales generadas**	**Frases con sentido**	**Frases nuevas con sentido (no en el corpus)**	**Frases con sentido incluidas en el corpus**
IN	HanaNLG$_{N-gram}$	140	51,43%	34,29%	17,14%
	HanaNLG$_{FLM}$ *WP*	21	33,33%	28,57%	4,76%
	HanaNLG$_{FLM}$ *LP*	33	75,75%	72,72%	3,03%
ES	HanaNLG$_{N-gram}$	95	56,84%	31,58%	25,26%
	HanaNLG$_{FLM}$ *WP*	67	77,61%	53,73%	23,88%
	HanaNLG$_{FLM}$ *LP*	64	79,69%	54,69%	25%

Table A.3: Resultados y comparación de los MLF empleados en HanaNLG$_{FLM}$ con respecto a HanaNLG$_{N-gram}$ para el escenario de tecnologías de asistencia. IN y ES se refieren al idioma, que en este caso es inglés y español respectivamente.

Como se observa en la tabla, los resultados obtenidos por cualquiera de las dos configuraciones de HanaNLG$_{FLM}$ superan casi todos los resultados de HanaNLG$_{N-gram}$, verificando su idoneidad. En el caso del español, los resultados de HanaNLG$_{FLM}$ han mejorado con respecto a los de HanaNLG$_{N-gram}$. Sin embargo, los de inglés no parecen mejorar mucho. La razón es que la mayoría de las frases generadas en inglés son realmente largas y pueden carecer de significado debido a errores gramaticales. Además, el número de frases generadas en inglés ha disminuido en comparación con las generadas en español. Esto se debe a que, cuando se selecciona la primera palabra de la oración, ésta es determina en base al *token* (<s>) y en la etiqueta gramatical más probable que aparece después de este *token*. En el caso del inglés, la etiqueta gramatical más probable es un pronombre, por lo tanto, se generan muy pocas frases que empiecen con un pronombre y terminen con un punto.

- **En el caso de que los modelos de lenguaje factorizados funcionen mejor que los modelos n-gramas, ¿qué tipo de factores proporcionan una mayor flexibilidad cuando se genera lenguaje, con respecto al contenido?**

  El objetivo principal de esta experimentación es probar varios factores para los MLF utilizados durante la generación y determinar cuál puede proporcionar más flexibilidad en términos de contenido.

  Para la generación de las frases de este experimento se utilizó el enfoque HanaNLG$_{FLM}$ con la inclusión de un ranking. Por lo tanto, este enfoque se basa en técnicas de *over-generation* y *ranking*. Este *ranking* es el que se usa en el módulo de *ranking* de frases de HanaNLG. Nos referiremos a este enfoque como HanaNLG$_{FLMR}$.

  En esta experimentación no se utilizó ninguna característica semilla para generar las frases (en vez de ellas se emplearon las palabras más frecuentes de los modelos de lenguaje), pero sí se emplearon diferentes factores para la generación de las mismas. Específicamente, se probaron las siguientes combinaciones de factores: (i) Palabra + Etiqueta gramatical (*WP*); (ii) Lema + Etiqueta gramatical (*LP*) y (iii) Synset + Etiqueta gramatical (*SP*). Para cada una de estas combinaciones se generaron 20 frases, usando como corpus de entrada una colección de 779 cuentos infantiles.

  La tabla A.4 resume los datos obtenidos de la evaluación manual realizada. En esta evaluación se evaluaron, por medio de cuestionarios que usa escalas de Likert de 5 puntos, los siguientes aspectos: (i) coherencia; (ii) Utilidad; (iii) errores gramaticales; y (iv) estructura.

Factores	**Coherencia**	**Utilidad**	**Errores gramaticales**	**Estructura**
HanaNLG$_{FLMR}$ *WP*	2.68	2,80	2,83	3,22
HanaNLG$_{FLMR}$ *LP*	3,08	3,31	3,00	3,53
HanaNLG$_{FLMR}$ *SP*	2,85	3,02	3,08	3,53

Table A.4: Resultados de la media de la escala Likert de 5 puntos con respecto a la coherencia, utilidad de la frase, errores gramaticales y estructura, de las frases generadas con MLF.

  La tabla muestra que los resultados obtenidos al emplear ambas configuraciones, *LP* y *SP*, superan a los de la configuración *WP* para cada uno de los criterios evaluados. Estos resultados muestran que el uso de factores más abstractos puede conducir a modelos con más potencial y expresividad. Esto demuestra que este tipo de factores aumentan la flexibilidad del lenguaje generado en términos de contenido, ya que permite la elección de distintos términos para un mismo lema o synset dependiendo del contexto para el que se esté generando.

- **¿El uso de características semilla junto con los modelos de lenguaje factorizados permite la generación de lenguaje para diferentes dominios y lenguajes?**

  El uso de las características semilla y MLF individualmente han dado buenos resultados en los experimentos anteriores. Además, su combinación también ha obtenido resultados prometedores en el caso de un solo dominio. Por lo tanto, el objetivo principal de esta experimentación es probar el rendimiento de esta combinación en diferentes dominios y también en diferentes idiomas.

  Para llevar a cabo este análisis emplearemos el enfoque HanaNLG$_{FLMR}$. Como no hay ningún cambio en la técnica para generar automáticamente el lenguaje, en esta experimentación, también nos referiremos al enfoque como HanaNLG$_{FLMR}$.

  En esta experimentación se analizaron dos escenarios distintos: (i) GLN para tecnologías de asistencia y (ii) GLN para frases con opinión. En el caso del primer escenario, se emplearon fonemas como característica semilla, mientras que en el segundo escenario, se usaron polaridades (positivo o negativo). Además, para cada uno de estos escenarios, se generaron frases tanto en español como en inglés. Para evaluar las frases generadas en esta experimentación se tuvieron en cuenta distintos aspectos para considerar si una frase tiene sentido o no: (i) si la frase tiene sentido por sí misma; ii) si la frase obtiene sentido al agregar algunos signos de puntuación; y, iii) si la frase adquiere significado al insertar una preposición que usualmente sigue al verbo principal.

  La tabla A.5 resume los resultados obtenidos de la evaluación manual. Como se observa en la tabla, los resultados obtenidos por HanaNLG$_{FLMR}$ son prometedores. Sin embargo, dado que el principal factor empleado dentro de la formación de MLF es el lema, las frases generadas no contienen palabras flexionadas. Además, como resultado de la evaluación realizada, las frases resultantes pueden no ser estrictamente correctas y pueden contener algunos errores.

- **¿En qué medida la integración de un módulo de flexión mejora la naturalidad y la expresividad del lenguaje generado?**

  La flexión morfológica es clave para que el lenguaje sea lo más natural posible. En este sentido, la naturalidad y expresividad del lenguaje generado en un sistema NLG puede mejorarse enriqueciendo el lenguaje a través de su morfología. Por lo tanto, el objetivo principal de esta experimentación es probar el rendimiento del módulo de flexión dentro de HanaNLG.

Escenario		**Frases generadas con sentido**	**Frases nuevas con sentido (no en el corpus)**	**Frases con sentido con característica semilla**
IN	GLN para tecnologías de asistencia	95%	70%	82,5%
	GLN para frases con opinión	100%	50%	50%
ES	GLN para tecnologías de asistencia	88,89%	40,74%	88,89%
	GLN para frases con opinión	100%	100%	100%

Table A.5: Tabla comparativa de las frases generadas por HanaNLG$_{FLMR}$ para los dos escenarios propuestos. IN y ES se refieren al idioma, que en este caso es inglés y español respectivamente.

Para lograr este objetivo, utilizaremos el módulo de flexión propuesto, configurándolo para el español, junto con el enfoque HanaNLG$_{FLMR}$. Además de esto, también queríamos probar la generación de frases relacionadas. Por lo tanto, se utiliza una gramática simple, basada en la estructura sujeto-verbo-objeto, para garantizar que algunos elementos de la frase aparezcan. Nos referiremos a este enfoque como HanaNLG$_{INF}$.

En esta experimentación se probó la generación de frases con dos tipos configuraciones distintas de flexión: (i) aleatoria y (ii) fija. En la primera configuración, se asigna un tiempo verbal aleatorio a cada una de las frases que componen el conjunto de frases. En el caso de la segunda configuración, el tiempo verbal para todo el conjunto de frases se fija en un tiempo verbal único, como el presente indicativo.

Tipo de fleción	**Coherencia**		**Errores gram.**		**Post-edición**	
	Media	Moda	Media	Moda	Media	Moda
HanaNLG$_{FLMR}$ LP (Sin)	2,65*	2	2,73*	3	2,75*	3
HanaNLG$_{INF}$ Fija	**3,36***	3	**3,57***	3	**3,54***	4
HanaNLG$_{INF}$ Aleatoria	3,31*	5	3,51*	4	3,48*	4

Table A.6: Resultados de las medias y las modas de la escala de Likert de 5 puntos con respecto a la coherencia, errores gramaticales y la facilidad de corrección de las frases generadas con flexión. * denota significancia con $p < 0.01$.

La tabla A.6 muestra las medias obtenidas de la evaluación manual. En esta evaluación se valoraron las frases generadas, mediante el uso de cuestionarios con escalas de Likert de 5 puntos, los siguientes aspectos: (i) coherencia; (ii) errores gramaticales; y (iii) post-edición (facilidad de corrección). Como se esperaba, ambas configuraciones de flexión logran mejores resultados para cada uno de los aspectos de la evaluación en comparación con no flexionar la frase. Estos resultados indican que la calidad de las frases mejoró con la inclusión del módulo de flexión.

- **¿La combinación de información semántica y sintáctica afecta la calidad del texto generado?**

  En los experimentos anteriores, hemos visto la experimentación realizada para verificar la idoneidad de las metodologías utilizadas en HanaNLG. Una vez que esto ha sido probado, ahora podemos enfocarnos en la evaluación del desempeño de HanaNLG como un enfoque completo. Por lo tanto, analizaremos si la inclusión de información semántica y sintáctica puede aumentar la calidad del texto generado.

  Para realizar este análisis, evaluamos el enfoque para generar texto en inglés en dos escenarios distintos: (i) GLN para tecnologías de asistencia y (ii) GLN para frases con opinión. Estos escenarios fueron escogidos porque vamos a comparar los resultados de HanaNLG con los obtenidos por HanaNLG$_{FLMR}$. Esto se debe a que en el campo de investigación de la GLN no hay *gold-standards* para poder comparar nuestra salida y, por lo que sabemos, no existen otros sistemas de realización que trabajen en estos escenarios específicos. Por lo tanto, es sumamente difícil compararnos con ningún sistema del estado de la cuestión.

  El objetivo principal de esta experimentación es generar, en el caso del primer escenario, una frase para cada uno de los fonemas ingleses (es decir, el inglés tiene un total de 44 fonemas) y una frase para cada polaridad (es decir, negativa y positiva) en el caso del segundo escenario.

Enfoque	**Escenario**	**Frases con sentido**	**Nuevas frases generadas**	**Frases con sentido y carac. semilla**
HanaNLG	GLN para tecnologías de asistencia	**97,73%**	**100%**	**100%**
	GLN para frases con opinión	**100%**	**100%**	**100%**
HanaNLG$_{FLMR}$*	GLN para tecnologías de asistencia	95%	70%	82,5%
	GLN para frases con opinión	100%	50%	50%

Table A.7: Resultados de la evaluación manual de HanaNLG y HanaNLG$_{FLMR}$ para los dos escenarios propuestos. *Estos resultados corresponden a los de la tabla A.5.

La tabla 5.13 resume los resultados obtenidos de la evaluación manual de HanaNLG en comparación con los de HanaNLG$_{FLMR}$. Como se ve en la tabla, HanaNLG logra mejores resultados en la generación de texto en ambos escenarios. En este sentido, estos resultados mejoran cada uno de los aspectos evaluados, generando lenguaje de nueva creación (original) con sentido (es decir, que no existe explícitamente en el corpus de entrada) y todas las frases generadas contienen palabras relacionadas con su respectiva característica semilla de entrada. En términos de contenido, el texto producido por nuestro enfoque contiene más información semántica gracias al uso de recursos como VerbNet y WordNet que nos dan un mayor control sobre el contenido generado. Los *frames* obtenidos de estos recursos nos proporcionan una estructura para la frase, eliminando la necesidad de tener una gramática compleja o costosa computacionalmente. Además, cabe mencionar que las frases generadas por HanaNLG han sido flexionadas automáticamente en pasado simple, dotándolas de una mayor naturalidad.

A la luz de estos resultados, se ha demostrado que una perspectiva híbrida para la GLN puede proporcionar más flexibilidad y calidad al lenguaje generado, permitiendo la adaptación del proceso de generación a diferentes propósitos y escenarios.

A.7.2 Evaluación extrínseca de HanaNLG

Además de la evaluación intrínseca de HanaNLG, su adaptación a otras tareas del PLN sirve para evaluar también de manera extrínseca nuestro enfoque. Específicamente, HanaNLG se ha probado en dos aplicaciones distintas de la tarea de generación automática de resúmenes: (i) generación de titulares y (ii) generación de líneas temporales. A continuación se detallan los resultados más significativos que se han obtenido.

- **Generación de titulares**

  La finalidad de esta tarea es la generación de un titular, a partir de una noticia de entrada, que describa el contenido de la misma. Por lo tanto, esta tarea se ha tratado tradicionalmente como generación de resúmenes automáticos de un solo documento. Para poder generar titulares con HanaNLG, se ha utilizado la característica semilla de "tópicos o palabras importantes". Esta característica semilla permite la generación de texto relacionado con un tópico o tema específico. En consecuencia, este tipo de característica semilla es adecuada para la tarea de generación de titulares.

  Para extraer los tópicos o palabras importantes de una noticia se emplearon las siguientes heurísticas del campo de la generación automática de resúmenes: (i) entidades nombradas (Named Entities: NE); (ii) asignación latente de Dirichlet (Latent Dirichlet Allocation: LDA); (iii) frecuencia de término (Term frequency: TF); y (iv) frecuencia de término-frecuencia inversa de frase (Term frecuency-Inverse sentence frecuency:

TF-ISF). Para la experimentación llevada a cabo, se empleraon los conjuntos de datos de la primera tarea del DUC 2003 y 2004, debido a que estas tareas estaban enfocadas a la generación de resúmenes muy cortos (10 palabras aproximádamente) de un solo documento, por lo que eran comparables con un titular. Para cada una de las noticias de estos conjuntos de datos, se generó un titular diferente, generando un total de 2496 titulares (624 titulares por cada una de las heurísticas) y 2000 titulares (500 titulares por cada una de las heurísticas) utilizando los documentos contenidos en los conjuntos de datos del DUC 2003 y 2004, respectivamente.

	DUC 2003		
Sistema	**R1 Cobertura**	**R1 Precisión**	**R1 Medida F**
COMPENDIUM	13,71	10,56	11,84
BestDUC03	**28,27**	**29,96**	**28.85**
LeadBaselineDUC03	19,18	25,04	21,31
HanaNLG-NE	23,82	28,17	25,61

Table A.8: Resultados de ROUGE del conjunto de datos del DUC 2003 para los sistemas competitivos y *HanaNLG*. Estos resultados se refieren la cobertura, la precisión y la medida F de ROUGE R1.

	DUC 2004		
Sistema	**R1 Cobertura**	**R1 Precisión**	**R1 Medida F**
Tan17	**28,97**	-	-
Takase16	28,80	-	-
Chopra16	28,68	-	-
Rush15	26,55	-	-
COMPENDIUM	14,08	12,50	13,12
BestDUC04	25,65	**27,36**	**26,26**
LeadBaselineDUC04	22,25	23,74	22,83
HanaNLG-NE	22,62	27,24	24,52

Table A.9: Resultados de ROUGE del conjunto de datos del DUC 2004 para los sistemas competitivos y *HanaNLG*. Estos resultados se refieren a la cobertura, la precisión y la medida F de ROUGE R1.

Para la evalución automática de estos titulares se utilizará la métrica ROUGE. Los resultados de ROUGE de los titulares generados utilizando la heurística NE en comparación con sistemas competitivos de generación automática de resúmenes se muestran en la tabla A.8 y en la tabla A.9. Aunque HanaNLG no supera a los mejores sistemas competitivos, éste ofrece resultados comparables, teniendo en cuenta que no fue diseñado en un principio como un sistema de generación de resúmenes. Con respecto a los *baselines* definidos para cada una de las tareas del DUC, HanaNLG supera sus resultados. Además, obtenemos mejores resultados que el sistema COMPENDIUM. Aunque COMPENDIUM es un sistema de resumen extractivo y genérico, extraer la frase más importante del documento no es suficiente para poder considerarlo un titular.

- **Generación de líneas temporales**

  Además de para la tarea de generación de titulares, HanaNLG se ha adaptado para la tarea de generación de líneas temporales. El objetivo de esta tarea es la creación de resúmenes narrativos basados en un orden natural de eventos en el tiempo (línea temporal), a partir de un conjunto de documentos de partida. Para poder generar este tipo de resúmenes con HanaNLG, el enfoque parte de un conjunto de eventos obtenidos a través de un sistema de Extracción de Líneas Temporales Enriquecidas (Navarro-Colorado & Saquete, 2016). El contenido de estos eventos es utilizado entonces por HanaNLG para generar frases relacionadas con los acontecimientos ocurridos en dichos eventos.

  Para esta experimentación se empleó como corpus el conjunto de datos de prueba de la tarea 4 del SemEval 2015[4]. Para cada una de las entidades contenidas en los documentos de dicho conjunto de datos se generó un resumen. Para generar estos resúmenes, se consideraron dos configuraciones distintas: (i) experimento *gold-standard* y (ii) experimento global. En el primer experimento se utilizaron las líneas temporales *gold-standard* provistas en la tarea 4 de SemEval, mientras que en el segundo experimento se utilizó el sistema de Extracción de Líneas Temporales Enriquecidas para obtener un esquema de línea temporal a partir de los documentos de SemEval.

	ROUGE-1			**ROUGE-2**			**ROUGE-L**			**ROUGE-SU4**		
	R	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
COMPENDIUM	0,317	0,370	0,312	0,114	0,154	0,121	0,296	0,348	0,293	0,142	0,180	0,145
GRAFENO	0,285	0,415	0,295	0,102	0,199	0,118	0,261	0,384	0,272	0,127	0,140	0,139
OTS	0,305	0,362	0,303	0,106	0,148	0,114	0,280	0,335	0,280	0,133	0,173	0,138
FirstEvent	0,323	0,583	0,402	0,141	0,270	0,179	0,316	0,570	0,392	0,140	0,264	0,176
LongestEvent	0,351	0,688	0,445	0,166	0,335	0,215	0,340	0,665	0,431	0,165	0,339	0,214
HanaNLG	**0,576**	**0,735**	**0,637**	**0,420**	**0,544**	**0,467**	**0,559**	**0,714**	**0,619**	**0,400**	**0,518**	**0,445**

Table A.10: Valores medios para la cobertura (R), la precisión (P) y la medida F (F) del experimento *gold-standard*. Comparación entre diferentes enfoques de generación de resúmenes y *baselines*.

Las tablas A.10 y A.11 muestran los resultados promedios de *recall* (R), *precision* (P) y *F1-measure* (F) para las siguientes métricas de ROUGE: (i) ROUGE-1, (ii) ROUGE-2, (iii) ROUGE-L y (iv) ROUGE-SU4. Como puede verse en las tablas, en ambos experimentos, superamos a los sistemas

[4] http://alt.qcri.org/semeval2015/task4/index.php?id=data

	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4		
	R	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**	**R**	**P**	**F**
COMPENDIUM	0,317	0,370	0,312	0,114	0,154	0,121	0,296	0,348	0,293	0,142	0,180	0,145
GRAFENO	0,285	0,415	0,295	0,102	0,199	0,118	0,261	0,384	0,272	0,127	0,140	0,139
OTS	0,305	0,362	0,303	0,106	0,148	0,114	0,280	0,335	0,280	0,133	0,173	0,138
FirstEvent	0,258	0,463	0,302	0,083	0,164	0,101	0,250	0,444	0,293	0,100	0,194	0,119
LongestEvent	0,251	0,524	0,312	0,088	0,196	0,114	0,245	0,510	0,305	0,099	0,225	0,125
HanaNLG	**0,433**	**0,595**	**0,470**	**0,263**	**0,363**	**0,284**	**0,422**	**0,579**	**0,457**	**0,260**	**0,360**	**0,282**

Table A.11: Valores medios para la cobertura (R), la precisión (P) y la medida F (F) del experimento global. Comparación entre diferentes enfoques de generación de resúmenes y *baselines* en un escenario real.

restantes. Esto indica que la combinación de técnicas de GLN con la identificación de eventos y la extracción de información temporal mejora la generación de resúmenes narrativos. Sin embargo, los resultados obtenidos para el experimento global son más bajos que los del experimento *gold-standard*. Esto se debe a que el sistema de Extracción de Líneas Temporales Enriquecidas puede introducir errores al extraer los eventos de los documentos.

A.8 Conclusiones y trabajo en progreso

Esta tesis se centra en el estudio de la fase de realización, dentro de la tarea de GLN, desde una perspectiva híbrida. Durante el desarrollo de esta tesis doctoral se han obtenido las siguientes publicaciones científicas: 4 artículos de revista (2 de ellos indexados en el Journal Citation Reports, en el primer y tercer cuartil; y los restantes indexados en Scopus); y 14 publicaciones en conferencias y talleres (incluyendo muchas conferencias prestigiosas del área de la GLN y del PLN como INLG, ENLG, CICLING, NLDB o SEPLN). A continuación se exponen las principales conclusiones y contribuciones científicas que aporta esta tesis, que se pueden resumir en los siguientes puntos:

- **Análisis del estado de la cuestión en lo que se refiere a los enfoques y metodologías para la generación del lenguaje natural.** Del extenso análisis llevado a cabo, hemos sido capaces de discernir algunas pistas sobre las direcciones del campo de la GLN. Esto esta relacionado con la necesidad de diseñar y desarrollar enfoques más flexibles y versátiles. En este sentido, debido a que los sistemas existentes están diseñados actualmente para propósitos, dominios y lenguajes muy concretos; su adaptación a otros dominios suele ser bastante costosa. Por lo tanto, la investigación en enfoques más fácilmente adaptables y flexibles supondría un gran avance en este área de investigación. Con respecto a los enfoques utilizados para el desarrollo de sistemas de GLN, los más usados son aquellos basados en conocimiento pero también aquellos que emplean técnicas estadísticas. Recientemente esta incrementando el uso de técnicas de aprendizaje profundo, sin embargo, no se han probado lo suficiente y el texto generado por alguno de ellos puede contener información errónea.

- **Análisis del estado de la cuestión en la evaluación de lenguaje automáticamente generado.** A pesar de las metodologías existentes de evaluación (automática o manual), la evaluación de los textos generados por enfoques de GLN sigue siendo un desafío abierto. Cuando hablamos de evaluación automática en GLN, ésta se lleva a cabo usando métricas de otras áreas de PLN. Sin embargo, estas métricas no pueden garntizar que el texto tengo sentido o que su estructura gramatical sea completamente correcta. Además, dado que en GLN no hay una única salida correcta y que hay una escasez de *gold-standards*, esto hace que la evaluación manual sea preferible en algunos casos. La evaluación manual suele realizarse colaborativamente y a través del uso de cuestionarios. Sin embargo, la principal preocupación de este tipo de evaluación es la subjetividad de los evaluadores, lo que lleva a una gran variedad en las valoraciones. Independientemente del tipo de metodología de evaluación, no hay un consenso sobre cuál usar, por lo que cada investigador decide cuál usar de acuerdo a sus necesidades.

- **Propuesta y desarollo de HanaNLG**. A la luz de las limitaciones detectadas a través del análisis del estado de la cuestión de la GLN, en esta tesis doctoral se ha propuesto, diseñado y desarrollado HanaNLG, el cuál es un enfoque híbrido basado en la fase de realización. Este enfoque es capaz de adaptar fácilmente el lenguaje generado a diferentes escenarios y dominios. Esto se realiza mediante el uso de características semillas y la aplicación de técnicas híbridas (es decir, la combinación de información basada en conocimiento —VerbNet y WordNet— y en modelos estadísticos —MLF—). La arquitectura de HanaNLG está compuesta por seis módulos diferentes. Los módulos de *Preprocesamiento* y *Selección de vocabulario* se ocupan del procesamiento de la entrada, el entrenamieno de los MLF usados durante el proceso de generación y la selección de vocabulario que compondrá el texto final. Los últimos cuatro módulos — *Generación de frases, Ranking de frases, Flexión de frases* y *Agregación de frases* — son responsables de la generación de la salida del enfoque. La salida será una frase o conjunto de frases flexionadas y bien estructuradas.

- **Evaluación intrínseca de HanaNLG**. Para verificar la idoneidad de las técnicas usadas durante el desarollo de HanaNLG así como las usadas en el enfoque completo, hemos realizado una evaluación incremental. En este sentido, este tipo de evaluación permitió la valoración de cada aspecto individual en el desarrollo de HanaNLG. Primero, analizando la adecuación de los modelos de lenguaje utilizados y el uso de características semillas en la generación del lenguaje. Después, analizando la flexibilidad de adaptación para la generación en diferentes escenarios. Y, finalmente, evaluando HanaNLG como un enfoque híbrido completo de realización En esta evaluación hemos obtenido buenos resultados en la generación de frases que son fácilmente adaptables a diferentes escenarios y propósitos. Concretamente, HanaNLG fue probado en los escenarios de GLN para tecnologías de asistencia y GLN para frases con opinión, teniendo sentido el 97,73% y el 100% de las frases generadas e incluyendo la característica semilla en cada uno de los escenarios, respectivamente. Además, con la inclusión de un módulo de flexión en el enfoque propuesto y la combinación de recursos semánticos y modelos estadísticos, la calidad del lenguaje generado ha mejorado. En este contexto, el número de frases con sentido (es decir, el 97,73% de las frases), con respecto a la versión de HanaNLG donde no se usaba información semántica (en la que el 95% de las frases generadas tenían sentido y el 82,5% tenían característica semilla), ha aumentado así como el número de frases nuevas con sentido que incluyen características semilla (es decir, el 100% de las frases). Asímismo, a partir de la extensa evaluación realizada, se pueden sacar algunas conclusiones sobre las ventajas y limitaciones de HanaNLG. En cuanto a sus ventajas, HanaNLG ha demostrado ser capaz de adaptar fácilmente el lenguaje generado a distintos dominios o propósitos gracias al uso de características semilla. Además de esto , la combinación de recursos semánticos e información estadística junto con las características semillas ha demostrado que aumenta la flexibilidad del lenguaje generado en términos de vocabulario. En contraste, su mayor limitación reside en dos cuestiones. En primer lugar, las frases generadas son cortas debido a la estructura obtenida de los *frames* usados durante el proceso de generación. En segundo lugar, algunos de los recursos empleados en el desarrollo del enfoque propuesto son dependientes del idioma. Por lo tanto, otros recursos específicos para un idioma objetivo podrían ser necesarios para poder generar lenguaje para ese idioma. Estas cuestiones formarán parte de las futuras líneas de investigación que podrán beneficiar y mejorar HanaNLG.

- **Evaluación extrínseca de HanaNLG como una aplicación para la tarea de generación automática de resúmenes**. Dado que el lenguaje generado por HanaNLG ha dado buenos resultados en deferentes escenarios, es importante analizar la adaptabilidad de este enfoque a otros campos del PLN. En particular, hemos enfocado este análisis al área de generación automática de resúmenes. Dentro de éste área, hemos probado HanaNLG bajo dos aplicaciones diferentes: generación de titulares y generación de líneas temporales. Dentro de la primera aplicación, HanaNLG fue usado para generar resúmenes abstractivos de un solo documento que resumen la idea principal de un noticia en la forma de un titular. En el caso de la segunda aplicación, hemos generado resúmenes de líneas temporales de varios documentos con el uso de un módulo externo de Extracción de Líneas Temporales Enriquecidas que ha sido integrado en HanaNLG. Estos resúmenes contienen eventos de una entidad específica que aparece en varios documentos y se presentan de una forma ordenada. Finalmente, de los resultados de ambas aplicaciones, cabe destacar que la adaptación de HanaNLG para la tarea de generación de titulares, aunque los resultados obtenidos no superan los de los sistemas competitivos, éstos sobrepasan los resultados de otros sistemas enfocados a generación de resúmenes. Además, la integración de HanaNLG, en el caso de la generación de líneas temporales de varios documentos, mejoran los resultados obtenidos sin afectar al rendimiento del sistema completo.

Como trabajos que se están llevando a cabo actualmente como continuación de esta tesis, y los que se pretenden realizar en un futuro, podemos destacar las siguientes líneas de investigación a corto, medio y largo plazo:

- **Análisis de HanaNLG en otro escenario**. Con el fin de amplir lo escenarios explorados para probar HanaNLG, actualmente estamos analizando el desempeño de HanaNLG en otros escenarios. Concretamente, este escenario está enfocado en la generación de cuentos infantiles dados algunos personajes y acciones. El texto generado en este escenario podría ser útil en la creación de nuevos cuentos infantiles basados en los gustos de los usuarios. Por lo tanto, con el objetivo a largo plazo de crear este tipo de cuentos infantiles, como un primer paso, hemos analizado HanaNLG en el contexto de generación de cuentos y además hemos propuesto un método para identificar automáticamente personajes en narrativas de ficción.
- **Análisis de otro tipo de recursos multilingües para la generación de texto en diferentes idiomas**. En este momento, la generación de texto en HanaNLG sólo es posibe para el inglés, en el caso del enfoque completo, debido a que algunos de los recursos usados en el desarrollo de nuestro enfoque son dependientes del lenguaje. Por lo tanto, la investigación y el análisis de otros recursos que nos permitan la generación de textos en diferentes idiomas es esencial. Estos recursos necesitarían contener tanto información lingüística como semántica al igual que VerbNet.
- **Análisis e investigación en enfoques de aprendizaje profundo para su inclusión en HanaNLG**. El uso de enfoques de aprendizaje profundo se ha incrementado en los últimos años en el campo del PLN. De la misma manera, algunos enfoques de GLN que utilizan este tipo de técnicas están emergiendo últimamente. Sin embargo, en el caso de la GLN, éstos no están lo sificientemente extendidos en dicho campo. Por lo tanto, queremos analizar si su inclusión en nuestro enfoque puede mejorar los resultados obtenidos y la calidad del

lenguaje generado. Este tipo de técnicas serían introducidas en los módulos de *Preprocesamiento* y *Generación de frases* para ayudar en la creación de los modelos de lenguaje y también en el proceso de generación.

– **Análisis e investigación de frases más largas y complejas**. En el estado actual de HanaNLG, la longitud de las frases generas es corta debido a la estructura de los *frames* usados. Por tanto, existe una necesidad de investigar cómo mejorar el lenguaje generado con respecto a su longitud y complejidad. En este sentido, la investigación de recursos que contengan información (por ejemplo, información más detallada sobre los componentes de una oración) que nos permita adaptar nuestro enfoque para la producción de frases más largas y con una estructura compleja (por ejemplo, incluyendo frases subordinadas) es esencial.

References

Ahlberg, M., Forsberg, M., & Hulden, M. (2014). Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 569–578).

Alnajjar, K., & Hämäläinen, M. (2018). A Master-Apprentice Approach to Automatic Creation of Culturally Satirical Movie Titles. In *Proceedings of the 11th International Conference on Natural Language Generation* (pp. 274–283). Association for Computational Linguistics.

Alonso, J. M., Ramos-Soto, A., Castiello, C., & Mencar, C. (2018). Explainable AI Beer Style Classifier. In *The SICSA Reasoning, Learning and Explainability Workshop 2018.*

Andonov, F., Slavova, V., & Petrov, G. (2016, 12). ON THE OPEN TEXT SUMMARIZER. *International Journal "Information Content and Processing"*, *3*(3), 278–287.

Androutsopoulos, I., Lampouras, G., & Galanis, D. (2013). Generating Natural Language Descriptions from OWL Ontologies: The Natural OWL System. *Journal Artificial Intelligence Research*, *48*(1), 671–715.

Angrosh, M., & Siddharthan, A. (2014). Text simplification using synchronous dependency grammars: Generalising automatically harvested rules . In *Proceedings of the 8th International Natural Language Generation Conference (INLG 2014)* (pp. 16–25). Association for Computational Linguistics.

Anselma, L., & Mazzei, A. (2018). Designing and testing the messages produced by a virtual dietitian. In *Proceedings of the 11th International Conference on Natural Language Generation* (pp. 244–253). Association for Computational Linguistics.

Arora, R., & Ravindran, B. (2008, Dec). Latent Dirichlet Allocation and Singular Value Decomposition Based Multi-document Summarization. In *2008 Eighth IEEE International Conference on Data Mining* (p. 713-718).

Aysolmaz, B., Leopold, H., Reijers, H. A., & Demirörs, O. (2018). A semi-automated approach for generating natural language requirements documents based on business process models. *Information and Software Technology*, *93*, 14 - 29.

Ballesteros, M., Mille, S., & Wanner, L. (2014). Classifiers for data-driven deep sentence generation. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)* (pp. 108–112). Association for Computational Linguistics.

Bangalore, S., & Rambow, O. (2000). Exploiting a Probabilistic Hierarchical Model for Generation. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1* (pp. 42–48). Association for Computational Linguistics.

Banko, M., Mittal, V. O., & Witbrock, M. J. (2000). Headline Generation Based on Statistical Translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (pp. 318–325). Association for Computational Linguistics.

Barros, C. (2016). *Aproximación Híbrida para la Generación del Lenguaje Natural.*

Barros, C. (2017). *Estudio de un enfoque híbrido para la Generación del Lenguaje Natural.*

Barros, C., Gkatzia, D., & Lloret, E. (2017a, September). Improving the Naturalness and Expressivity of Language Generation for Spanish. In *Proceedings of the 10th International Conference on Natural Language Generation* (pp. 41–50). Santiago de Compostela, Spain: Association for Computational Linguistics.

Barros, C., Gkatzia, D., & Lloret, E. (2017b). Inflection Generation for Spanish Verbs using Supervised Learning. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP* (pp. 136–141). Association for Computational Linguistics.

Barros, C., & Lloret, E. (2015a). *Aproximación Híbrida para la Generación del Lenguaje Natural.*

Barros, C., & Lloret, E. (2015b). Input Seed Features for Guiding the Generation Process: A Statistical Approach for Spanish. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)* (pp. 9–17). Association for Computational Linguistics.

Barros, C., & Lloret, E. (2015c). *Proposal of a Data-to-text Natural Language Generation Approach to Create Stories for Dyslalic Children.*

Barros, C., & Lloret, E. (2015d). *Statistical NLG based on phonemes: a preliminary algorithm.*

Barros, C., & Lloret, E. (2016). Generating sets of related sentences from input seed features. In *Proceedings of the 2nd International Workshop on Natu-*

ral Language Generation and the Semantic Web (WebNLG 2016) (pp. 1–4). Association for Computational Linguistics.

Barros, C., & Lloret, E. (2017). Analysing the influence of semantic knowledge in natural language generation. In *2017 Twelfth International Conference on Digital Information Management (ICDIM)* (p. 185-190).

Barros, C., & Lloret, E. (2017). A Multilingual Multi-domain Data-to-Text Natural Language Generation Approach. *Procesamiento del Lenguaje Natural*, *58*, 45–52.

Barros, C., & Lloret, E. (2018). Surface Realisation Using Factored Language Models and Input Seed Features. In F. Castro, S. Miranda-Jiménez, & M. González-Mendoza (Eds.), *Advances in Computational Intelligence. MICAI 2017* (pp. 15–26). Cham: Springer International Publishing.

Barros, C., & Lloret, E. (2019). *HanaNLG: A Flexible Hybrid Approach for Natural Language Generation.*

Barros, C., Vicente, M., & Lloret, E. (2019). *Tackling the Challenge of Computational Identification of Characters in Fictional Narratives.*

Barzilay, R., & Lapata, M. (2005). Collective content selection for concept-to-text generation. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 331–338).

Bateman, J., & Zoch, M. (2003). *Natural Language Generation.* Oxford University Press.

Belz, A., & Kow, E. (2010). Comparing Rating Scales and Preference Judgements in Language Evaluation. In *Proceedings of the 6th International Natural Language Generation Conference* (pp. 7–15). Association for Computational Linguistics.

Belz, A., & Reiter, E. (2006). Comparing Automatic and Human Evaluation of NLG Systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 313–320). The Association for Computer Linguistics.

Berger, A., & Lafferty, J. (2017). Information Retrieval As Statistical Translation. *SIGIR Forum*, *51*(2), 219–226.

Bernardi, R., Cakici, R., Elliott, D., Erdem, A., Erdem, E., Ikizler-Cinbis, N., Keller, F., Muscat, A., & Plank, B. (2016). Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures. *Journal of Artificial Intelligence Research*, *55*(1), 409–442.

Bernardos, M. d. S. (2003). *Marco metodológico para la construcción de sistemas de generación de lenguaje natural* (Unpublished doctoral dissertation).

Informatica.

Bernardos, M. d. S. (2007). ¿Qué es la generación de lenguaje natural? Una visión general sobre el proceso de generación. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, 11*(34), 105–128.

Bilmes, J. A., & Kirchhoff, K. (2003). Factored Language Models and Generalized Parallel Backoff. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* (pp. 4–6). Association for Computational Linguistics.

Binh Tran, G., Alrifai, M., & Quoc Nguyen, D. (2013). Predicting Relevant News Events for Timeline Summaries. In *Proceedings of the 22Nd International Conference on World Wide Web* (pp. 91–92). New York, NY, USA: ACM.

Binstead, K., & Ritchie, G. (1997). Computational rules for punning riddles. In *Humor* (Vol. 10, pp. 25–76). Walter de Gruyter & Co.

Bollmann, M. (2011). Adapting SimpleNLG to German. In *Proceedings of the 13th European Workshop on Natural Language Generation* (pp. 133–138). Association for Computational Linguistics.

Bouayad-Agha, N., Casamayor, G., Mille, S., & Wanner, L. (2012, August). Perspective-oriented Generation of Football Match Summaries: Old Tasks, New Challenges. *ACM Transactions on Speech and Language Processing, 9*(2), 3:1–3:31.

Brad, F., & Rebedea, T. (2017). Neural Paraphrase Generation using Transfer Learning. In *Proceedings of the 10th International Conference on Natural Language Generation* (pp. 257–261). Association for Computational Linguistics.

Calder, J., Evans, R., Mellish, C., & Reape, M. (1999). *"Free choice" and templates: how to get both at the same time* (Tech. Rep. No. D-99-01). Brighton: ITRI, University of Brighton.

Carter, R., & McCarthy, M. (2006). *Cambridge grammar of English: a comprehensive guide; spoken and written English grammar and usage.* Ernst Klett Sprachen.

Castro Ferreira, T., Moussallem, D., Kádár, Á., Wubben, S., & Krahmer, E. (2018). NeuralREG: An end-to-end approach to referring expression generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1959–1969). Association for Computational Linguistics.

Cawsey, A. J., Jones, R. B., & Pearson, J. (2000). The evaluation of a personalised health information system for patients with cancer. *User Modeling and User-Adapted Interaction, 10*(1), 47–72.

Charniak, E., & al., E. (2000). *BLLIP 1987-89 WSJ Corpus Release 1 LDC2000T43.* Philadelphia: Linguistic Data Consortium.

Cheng, H., Mellish, C. S., & O'Donnell, M. (1997). Aggregation based on text structure for descriptive text generation. In *Proceedings of the PhD Workshop on Natural Language Generation, 9th European Summer School in Logic, Language and Information* (pp. 18–22).

Chieu, H. L., & Lee, Y. K. (2004). Query Based Event Extraction Along a Timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 425–432). New York, NY, USA: ACM.

Chopra, S., Auli, M., & Rush, A. M. (2016). Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 93–98). Association for Computational Linguistics.

Cole, R. (Ed.). (1997). *Survey of the State of the Art in Human Language Technology.* New York, NY, USA: Cambridge University Press.

Colmenares, C. A., Litvak, M., Mantrach, A., & Silvestri, F. (2015). HEADS: Headline Generation as Sequence Prediction Using an Abstract Feature-Rich Space. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL (NAACL'15)* (pp. 133–142).

Colmenares, C. A., Litvak, M., Mantrach, A., Silvestri, F., & Rodríguez, H. (2019). Headline Generation as a Sequence Prediction with Conditional Random Fields. In *Multilingual Text Analysis* (p. 201-243). World Scientific.

Conde-Clemente, P., Alonso, J. M., & Trivino, G. (2018). Toward automatic generation of linguistic advice for saving energy at home. *Soft Computing*, *22*(2), 345–359.

Conde-Guzón, P., Quirós-Expósito, P., Conde-Guzón, M. J., & Bartolomé-Albistegui, M. T. (2014). Perfil neuropsicológico de niños con dislalias: alteraciones mnésicas y atencionales. *Anales de Psicología*, *30*, 1105 - 1114.

Conroy, J. M., Stewart, J. G., & Schlesinger, J. D. (2005). CLASSY Query-Based Multi-Document Summarization. In *In Proceedings of the Document Understanding Conf. Wksp. 2005 (DUC 2005) at the Human Language Technology Conf./Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP).*

Cruz, F. L., Troyano, J. A., Pontes, B., & Ortega, F. J. (2014). Building layered, multilingual sentiment lexicons at synset and lemma levels . *Expert Systems with Applications*, *41*(13), 5984 - 5994.

Dale, R., & White, M. (2007). Shared Tasks and Comparative Evaluation in Natural Language Generation. In *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation* (pp. 1–6).

Dalianis, H. (1996). Aggregation as a subtask of text and sentence planning. In *Proceedings of the 9th Florida Artificial Intelligence Research Symposium* (pp. 1–5).

Dalianis, H. (1999). Aggregation in Natural Language Generation. *Computational Intelligence, 15*(4), 384-414.

Davey, A. (1974). *The formalisation of discourse production* (PhD Thesis). University of Edinburgh.

Delicado, L., Sànchez, J., Carmona, J., & Padro, L. (2017, Sep). NLP4BPM : Natural language processing tools for business process management. In *International Conference on Business Process Management* (pp. 1–5).

de Rosis, F., & Grasso, F. (2000). Affective Natural Language Generation. In A. Paiva (Ed.), *Affective Interactions: Towards a New Generation of Computer Interfaces* (pp. 204–218). Springer Berlin Heidelberg.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research* (pp. 138–145). Morgan Kaufmann Publishers Inc.

Dorr, B., Zajic, D., & Schwartz, R. (2003). Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop* (pp. 1–8).

Durrett, G., & DeNero, J. (2013). Supervised Learning of Complete Morphological Paradigms. In *Proceedings of the North American Chapter of the Association for Computational Linguistics* (pp. 1185–1195).

Dzikovska, M. O., Isard, A., Bell, P., Moore, J. D., Steinhauser, N., & Campbell, G. (2011). Beetle II: An Adaptable Tutorial Dialogue System. In *Proceedings of the SIGDIAL 2011 Conference* (pp. 338–340).

Espinosa, D., Rajkumar, R., White, M., & Berleant, S. (2010). Further Meta-Evaluation of Broad-Coverage Surface Realization. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (pp. 564–574). Association for Computational Linguistics.

Evans, R., Piwek, P., & Cahill, L. (2002). What is NLG? In *Proceedings of Second International Natural Language Generation* (pp. 144–151).

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* MIT Press.

Fernández, J., Gutiérrez, Y., Gómez, J. M., Martínez-Barco, P., Montoyo, A., &

Muñoz, R. (2013). Sentiment Analysis of Spanish Tweets Using a Ranking Algorithm and Skipgrams. *Proc. of the TASS workshop at XXIX Conference of the Spanish Society for Natural Language Processing (SEPLN 2013)*, 133–142.

Ferres, L., Parush, A., Roberts, S., & Lindgaard, G. (2006). Helping people with visual impairments gain access to graphical information through natural language: The iGraph system. In *Proceedings of the 10th International Conference on Computers Helping People with Special Needs* (pp. 1122–1130). Springer.

Fiedler, A. (2005). Natural language proof explanation. In *Mechanizing Mathematical Reasoning* (pp. 342–363). Springer.

Filippova, K., Alfonseca, E., Colmenares, C. A., Kaiser, L., & Vinyals, O. (2015). Sentence Compression by Deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 360–368). Association for Computational Linguistics.

Finlayson, M. A. (2014). Java Libraries for Accessing the Princeton Wordnet: Comparison and Evaluation. In *Proceedings of the 7th International Global WordNet Conference (GWC 2014)* (pp. 78–85).

Forrest, J., Sripada, S., Pang, W., & Coghill, G. (2018). Towards making NLG a voice for interpretable Machine Learning. In *Proceedings of the 11th International Conference on Natural Language Generation* (pp. 177–182). Association for Computational Linguistics.

Frank, E., Hall, M. A., & Witten, I. H. (2016). *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"* (4th ed.). Morgan Kaufmann.

García Ibáñez, C., Hervás, R., & Gervás, P. (2004). Una arquitectura software para el desarrollo de aplicaciones de generación de lenguaje natural. *Procesamiento del Lenguaje Natural*, *33*, 111–118.

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 Internet of Accessible Things* (pp. 19:1–19:4). ACM.

Gardent, C., & Kruszewski, G. (2012, May). Generation for Grammar Engineering. In *INLG 2012 Proceedings of the Seventh International Natural Language Generation Conference* (pp. 31–39). Association for Computational Linguistics.

Gardent, C., & Perez-Beltrachini, L. (2017). A Statistical, Grammar-Based Approach to Microplanning . *Computational Linguistics*, *43*(1), 1–30.

Garoufi, K., & Koller, A. (2011). Combining symbolic and corpus-based approaches for the generation of successful referring expressions. In *Proceedings of the 13th European Workshop on Natural Language Generation* (pp. 121–131).

Gatt, A., & Krahmer, E. (2018, January). Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation. *Journal of Artificial Intelligence Research*, *61*(1), 65–170.

Gatt, A., Portet, F., Reiter, E., Hunter, J., Mahamood, S., Moncur, W., & Sripada, S. (2009). From Data to Text in the Neonatal Intensive Care Unit: Using NLG Technology for Decision Support and Information Management. *AI Communications*, *22*(3), 153–186.

Gatt, A., & Reiter, E. (2009). SimpleNLG: A Realisation Engine for Practical Applications. In *Proceedings of the 12th European Workshop on Natural Language Generation* (pp. 90–93). Association for Computational Linguistics.

Gatti, L., Ozbal, G., Guerini, M., Stock, O., & Strapparava, C. (2016). Heady-Lines: A Creative Generator Of Newspaper Headlines. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (pp. 79–83). ACM.

Ge, T., Pei, W., Ji, H., Li, S., Chang, B., & Sui, Z. (2015). Bring you to the past: Automatic Generation of Topically Relevant Event Chronicles. 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. 575–585). Association for Computational Linguistics.

Gervás, P. (2017). Template-Free Construction of Rhyming Poems with Thematic Cohesion. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)* (pp. 21–28). Association for Computational Linguistics.

Gkatzia, D., & Mahamood, S. (2015). A Snapshot of NLG Evaluation Practices 2005 - 2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)* (pp. 57–60). Association for Computational Linguistics.

Gkatzia, D., Rieser, V., & Lemon, O. (2016). How to talk to strangers: Generating medical reports for first-time users. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (p. 579-586).

Goldberg, E., Driedger, N., & Kittredge, R. I. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, *9*(2), 45–53.

Gong, J., Ren, W., & Zhang, P. (2017). An automatic generation method of sports news based on knowledge rules. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)* (p. 499-502).

Gottschall, J. (2012). *The Storytelling Animal.* Houghton Mifflin Harcourt.

Grosz, B. J., & Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational linguistics, 12*(3), 175–204.

Grosz, B. J., Weinstein, S., & Joshi, A. K. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational linguistics, 21*(2), 203–225.

Gupta, V., & Lehal, G. S. (2011, November). Article: Named Entity Recognition for Punjabi Language Text Summarization. *International Journal of Computer Applications, 33*(3), 28-32.

Gutiérrez Vázquez, Y., Fernández Orquín, A., Montoyo Guijarro, A., & Vázquez Pérez, S. (2011). Integración de recursos semánticos basados en WordNet. *Procesamiento del Lenguaje Natural, 47*, 161–168.

Halliday, M. A. (1985). *An Introduction to Functional Grammar.* Edward Arnold.

Haque, M. M., Pervin, S., & Begum, Z. (2015). Automatic Bengali news documents summarization by introducing sentence frequency and clustering. In *2015 18th International Conference on Computer and Information Technology (ICCIT)* (p. 156-160).

Hardy, H., & Vlachos, A. (2018). Guided Neural Language Generation for Abstractive Summarization using Abstract Meaning Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 768–773). Association for Computational Linguistics.

Hervás, R., & Gervás, P. (2008). Descripción de Entidades y Generación de Expresiones de Referencia en la Generación Automática de Discurso. *Procesamiento de Lenguaje Natural, 41*, 217-224.

Hovav, M. R., Doron, E., & Sichel, I. (2010). *Lexical Semantics, Syntax, and Event Structure.* Oxford: Oxford University Press.

Hovy, E. H. (1988). *Generating Natural Language Under Pragmatic Constraints.* L. Erlbaum Associates Inc.

Huang, C., Zaiane, O., Trabelsi, A., & Dziri, N. (2018). Automatic Dialogue Generation with Expressed Emotions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (pp. 49–54). Association for Computational Linguistics.

Hunter, J., Freer, Y., Gatt, A., Reiter, E., Sripada, S., & Sykes, C. (2012). Automatic Generation of Natural Language Nursing Shift Summaries in Neonatal Intensive Care: BT-Nurse. *Artificial Intelligence in Medicine, 56*(3), 157–172.

Isard, A., Brockmann, C., & Oberlander, J. (2006). Individuality and Alignment in Generated Dialogues. In *Proceedings of the INLG* (pp. 25–32). Association for Computational Linguistics.

Jacko, J. A. (2012). *Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, Third Edition* (3rd ed.). Boca Raton, FL, USA: CRC Press, Inc.

Ji, H., Grishman, R., Chen, Z., & Gupta, P. (2009). Cross-document Event Extraction and Tracking: Task, Evaluation, Techniques and Challenges. In *Proceedings of the International Conference RANLP-2009* (pp. 166–172). Association for Computational Linguistics.

Jing, H., & McKeown, K. R. (2000). Cut and Paste Based Text Summarization. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics.*

Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., & Maloor, P. (n.d.). MATCH: An architecture for multimodal dialogue systems..

Joshi, A. K., & Schabes, Y. (1997). Tree-adjoining grammars. In *Handbook of formal languages* (pp. 69–123). Springer.

Kantrowitz, M., & Bates, J. (1992). Integrated natural language generation systems. In *Proceedings of the 6th International Workshop on Natural Language Generation: Aspects of Automated Natural Language Generation* (pp. 13–28). Springer.

Kirchhoff, K., Bilmes, J., & Duh, K. (2007). *Factored language models tutorial.*

Koller, A., Byron, D., Cassell, J., Dale, R., Moore, J., Oberlander, J., & Striegnitz, K. (2009). The software architecture for the first challenge on generating instructions in virtual environments. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session* (pp. 33–36).

Koller, A., & Petrick, R. (2011). Experiences with planning for natural language generation. *Computational Intelligence*, *27*(1), 23–40.

Koller, A., & Stone, M. (2007). Sentence generation as a planning problem. In *ACL.*

Kondadadi, R., Howald, B., & Schilder, F. (2013). A Statistical NLG Framework for Aggregated Planning and Realization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1406–1415). Association for Computational Linguistics.

Konstas, I., & Lapata, M. (2013). A Global Model for Concept-to-Text Generation. *Journal of Artificial Intelligence Research (JAIR)*, *48*(1), 305–346.

Laclaustra, I. M., Ledesma, J. L., Méndez, G., & Gervás, P. (2014). Kill the Dragon and Rescue the Princess: Designing a Plan-Based Multi-agent Story Generator. In *5th International Conference on Computational Creativity* (pp. 347–350).

Langkilde, I., & Knight, K. (1998). Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1* (pp. 704–710). Association for Computational Linguistics.

Lavie, A., & Agarwal, A. (2007). Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation* (pp. 228–231). Association for Computational Linguistics.

Lavoie, B., & Rambow, O. (1997). A Fast and Portable Realizer for Text Generation Systems. In *Proceedings of the Fifth Conference on Applied Natural Language Processing* (pp. 265–268). Association for Computational Linguistics.

Lebret, R., Grangier, D., & Auli, M. (2016). Neural Text Generation from Structured Data with Application to the Biography Domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1203–1213). Association for Computational Linguistics.

Li, C., Xu, W., Li, S., & Gao, S. (2018). Guiding Generation for Abstractive Text Summarization Based on Key Information Guide Network . In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (pp. 55–60). Association for Computational Linguistics.

Lim-Cheng, N. R., Fabia, G. I. G., Quebral, M. E. G., & Yu, M. T. (2014). Shed: An Online Diet Counselling System. In *DLSU Research Congress 2014* (pp. 1–7).

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the Association for Computational Linguistics Workshop* (pp. 74–81).

Liu, B., Hu, M., & Cheng, J. (2005). Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web* (pp. 342–351).

Liu, M., Calvo, R. A., & Rus, V. (2012). G-Asks: An Intelligent Automatic Question Generation System for Academic Writing Support. *Dialogue and Discourse, 3*(2), 101–124.

Lloret, E., Boldrini, E., Vodolazova, T., Martínez-Barco, P., Muñoz, R., & Palomar, M. (2015). A novel concept-level approach for ultra-concise opinion summarization. *Expert Systems with Applications*, *42*(20), 7148–7156.

Lloret, E., & Palomar, M. (2013). COMPENDIUM: a text summarisation tool for generating summaries of multiple purposes, domains, and genres. *Natural Language Engineering*, *19*(2), 147–186.

Lobo, P. V., & De Matos, D. M. (2010). Fairy Tale Corpus Organization Using Latent Semantic Mapping and an Item-to-item Top-n Recommendation Algorithm. In *Language Resources and Evaluation Conference - LREC 2010, European Language Resources Association (ELRA).*

Macdonald, I., & Siddharthan, A. (2016). Summarising News Stories for Children. In *Proceedings of the 9th International Natural Language Generation conference* (pp. 1–10). Association for Computational Linguistics.

Mani, I. (1999). *Advances in Automatic Text Summarization* (M. T. Maybury, Ed.). Cambridge, MA, USA: MIT Press.

Mani, I., Pustejovsky, J., & Gaizauskas, R. (2005). *The Language of Time.* Oxford: Oxford University Press.

Manjavacas, E., Karsdorp, F., Burtenshaw, B., & Kestemont, M. (2017). Synthetic Literature: Writing Science Fiction in a Co-Creative Process. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)* (pp. 29–37). Association for Computational Linguistics.

Mann, W. (1999). *IntroducciÃşón a la TeorÃŋía de la Estructura Retórica (Rhetorical Structure Theory: RST).*

Mann, W., & Thompson, S. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse*, *8*(3), 243–281.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations* (pp. 55–60).

Mazzei, A., Battaglino, C., & Bosco, C. (2016). SimpleNLG-IT: adapting SimpleNLG to Italian. In *Proceedings of the 9th International Natural Language Generation conference* (pp. 184–192). Association for Computational Linguistics.

McDonald, D. D. (2010). Natural Language Generation. In *Handbook of natural language processing* (pp. 121–144). CRC Press.

Mel'cuk, I. A., et al. (1988). *Dependency syntax: theory and practice.* SUNY press.

Mellish, C., & Dale, R. (1998). Evaluation in the context of natural language generation. *Computer Speech & Language*, *12*(4), 349–373.

Mellish, C., Scott, D., Cahill, L., Paiva, D., Evans, R., & Reape, M. (2006). A reference architecture for natural language generation systems. *Natural language engineering*, *12*(01), 1–34.

Mille, S., Ballesteros, M., Burga, A., Casamayor, G., & Wanner, L. (2016). Multilingual Natural Language Generation within Abstractive Summarization. In *Proceedings of the 1st International Workshop on Multimodal Media Data Analytics co-located with the 22nd European Conference on Artificial Intelligence, MMDA@ECAI 2016* (pp. 33–38).

Mille, S., Burga, A., & Wanner, L. (2013). AnCora-UPF: A Multi-Level Annotation of Spanish. *DepLing 2013*, 217.

Mille, S., & Wanner, L. (2008). Multilingual summarization in practice: the case of patent claims. In *Proceedings of the 12th European Association of Machine Translation conference* (pp. 120–129).

Minard, A.-L., Speranza, M., Agirre, E., Aldabe, I., van Erp, M., Magnini, B., Rigau, G., & Urizar, R. (2015). SemEval-2015 Task 4: TimeLine: Cross-Document Event Ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation* (pp. 778–786). Association for Computational Linguistics.

Mitchell, M., Bohus, D., & Kamar, E. (2014, June). Crowdsourcing Language Generation Templates for Dialogue Systems. In *Proceedings of the INLG and SIGDIAL 2014 Joint Session* (pp. 172–180). Philadelphia, Pennsylvania, U.S.A.: Association for Computational Linguistics.

Mitchell, M., Han, X., Dodge, J., Mensch, A., Goyal, A., Berg, A., Yamaguchi, K., Berg, T., Stratos, K., & Daumé, H., III. (2012). Midge: Generating Image Descriptions from Computer Vision Detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 747–756). Stroudsburg, PA, USA: Association for Computational Linguistics.

Morales, J. L. O. (1992). *Nuevo método de ortografía*. Verbum.

Munigala, V., Mishra, A., Tamilselvam, S. G., Khare, S., Dasgupta, R., & Sankaran, A. (2018). PersuAIDE ! An Adaptive Persuasive Text Generation System for Fashion Domain. In *Companion Proceedings of the The Web Conference 2018* (pp. 335–342). International World Wide Web Conferences Steering Committee.

Natsum: Narrative abstractive summarization through cross-document timeline generation. (2019). *Information Processing & Management*, *56*(5), 1775 - 1793.

Navarro-Colorado, B., & Saquete, E. (2016). Cross-document event ordering through temporal, lexical and distributional knowledge. *Knowledge-Based Systems*, *110*, 244–254.

Nenkova, A., & McKeown, K. (2011). Automatic Summarization. *Foundations and Trends in Information Retrieval*, *5*(2-3), 103-233.

Nenkova, A., Vanderwende, L., & McKeown, K. (2006). A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors That Influence Summarization. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 573–580). New York, NY, USA: ACM.

Nesterenko, L. (2016). Building a System for Stock News Generation in Russian. In *Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web (WebNLG 2016)* (pp. 37–40). Association for Computational Linguistics.

Over, P., Dang, H., & Harman, D. (2007). DUC in Context. *Information Processing and Management: an International Journal*, *43*(6), 1506–1520.

Padró, L., & Stanilovsky, E. (2012). FreeLing 3.0: Towards Wider Multilinguality. In *Proceedings of the Eight International Conference on Language Resources and Evaluation* (pp. 2473–2479).

Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics* (pp. 271–278). Association for Computational Linguistics.

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.

Paris, C., Scott, D., Green, N., McCoy, K., & McDonald, D. (2007). Desiderata for Evaluation of Natural Language Generation. In *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation* (pp. 9–15).

Perez-Beltrachini, L., Gardent, C., & Kruszewski, G. (2012). Generating Grammar Exercises. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP* (pp. 147–156). Association for Computational Linguistics.

Pita Fernández, S. (1996). Determinación del tamaño muestral. *CAD ATEN PRIMARIA 1996*, *3*, 138–14.

Pittaras, N., Montanelli, S., Giannakopoulos, G., Ferrara, A., & Karkaletsis, V. (2019). Crowdsourcing in Single-document Summary Evaluation: The Argo

Way. In *Multilingual Text Analysis* (p. 245-280).

Radev, D., Allison, T., Blair-Goldensohn, S., Blitzer, J., Çelebi, A., Dimitrov, S., Drabek, E., Hakim, A., Lam, W., Liu, D., Otterbacher, J., Qi, H., Saggion, H., Teufel, S., Topper, M., Winkel, A., & Zhang, Z. (2004). MEAD - A Platform for Multidocument Multilingual Text Summarization. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04).* European Language Resources Association (ELRA).

Ramos-Soto, A., Bugarín, A. J., Barro, S., & Taboada, J. (2015). Linguistic Descriptions for Automatic Generation of Textual Short-Term Weather Forecasts on Real Prediction Data. *IEEE Transactions on Fuzzy Systems*, *23*(1), 44-57.

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 and Systems*, *285*, 31 - 51. (Special Issue on Linguistic Description of Time Series)

Ramos Soto, A., Janeiro Gallardo, J., & Bugarín Diz, A. (2017). Adapting SimpleNLG to Spanish. In *Proceedings of the 10th International Conference on Natural Language Generation* (pp. 144–148). Association for Computational Linguistics.

Randolph, J. J. (2008). *Online kappa calculator [Computer software]. Retrieved from http://justus.randolph.name/kappa.*

Řehůřek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45–50). Valletta, Malta: ELRA.

Reiter, E. (2007). An architecture for data-to-text systems. In *Proceedings of the Eleventh European Workshop on Natural Language Generation* (pp. 97–104).

Reiter, E. (2010). *Natural Language Generation.* Wiley-Blackbell.

Reiter, E., & Belz, A. (2009). An Investigation into the Validity of Some Metrics for Automatically Evaluating Natural Language Generation Systems. *Computational Linguistics*, *35*(4), 529–558.

Reiter, E., & Dale, R. (2000). *Building Natural Language Generation Systems.* Cambridge University Press.

Reiter, E., Robertson, R., & Osman, L. M. (2003). Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, *144*(1), 41–58.

Reiter, E., Sripada, S., Hunter, J., Yu, J., & Davy, I. (2005a). Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, *167*(1), 137–169.

Reiter, E., Sripada, S., Hunter, J., Yu, J., & Davy, I. (2005b, September). Choosing Words in Computer-generated Weather Forecasts. *Artif. Intell.*, *167*(1-2), 137–169.

Reiter, E., Turner, R., Alm, N., Black, R., Dempster, M., & Waller, A. (2009). Using NLG to help language-impaired users tell stories and participate in social dialogues. In *Proceedings of the 12th European Workshop on Natural Language Generation* (pp. 1–8). Association for Computational Linguistics.

Resnik, P., & Lin, J. (2010). Evaluation of NLP Systems. In *The Handbook of Computational Linguistics and Natural Language Processing* (pp. 271–295). Wiley-Blackwell.

Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here? In *Proceedings of the IEEE* (p. 1270-1278).

Rush, A. M., Chopra, S., & Weston, J. (2015). A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 379–389). Association for Computational Linguistics.

Rvachew, S., Rafaat, S., & Martin, M. (1999). Stimulability, Speech Perception Skills, and the Treatment of Phonological Disorders. *American Journal of Speech-Language Pathology*, *8*(1), 33-43.

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* (pp. 208–216). Association for Computational Linguistics.

Schuler, K. K. (2005). *Verbnet: A broad-coverage, comprehensive verb lexicon* (Unpublished doctoral dissertation).

Scott, D., & Moore, J. (2007). An NLG evaluation competition? Eight reasons to be cautious. In *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation* (pp. 22–23).

Sevilla, A. F., Fernandez-Isabel, A., & Díaz, A. (2016). Enriched semantic graphs for extractive text summarization. In *Conference of the Spanish Association for Artificial Intelligence* (pp. 217–226). Springer International Publishing.

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

Stolcke, A. (2002). SRILM - An Extensible Language Modeling Toolkit. In *Proceedings International Conference on Spoken Language Processing, vol 2.* (p. 901-904).

Subramanian, S., Rajeswar, S., Dutil, F., Pal, C., & Courville, A. (2017). Adversarial

Generation of Natural Language. In *Proceedings of the 2nd Workshop on Representation Learning for NLP* (pp. 241–251). Association for Computational Linguistics.

Takase, S., Suzuki, J., Okazaki, N., Hirao, T., & Nagata, M. (2016). Neural Headline Generation on Abstract Meaning Representation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1054–1059). Association for Computational Linguistics.

Tan, J., Wan, X., & Xiao, J. (2017). From Neural Sentence Summarization to Headline Generation: A Coarse-to-fine Approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (pp. 4109–4115). AAAI Press.

Tantuğ, A. C., & Adalı, E. (2018). "Machine Translation Between Turkic Languages". In K. Oflazer & M. Saraçlar (Eds.), *"Turkish Natural Language Processing"* (pp. 237–254). Cham: Springer International Publishing.

Theune, M., Hielkema, F., & Hendriks, P. (2006). Performing aggregation and ellipsis using discourse structures. *Research on Language and Computation, 4*(4), 353–375.

Tran, G. B., Tran, A. T., Tran, N.-K., Alrifai, M., & Kanhabua, N. (2013). Leverage Learning to rank in an optimization framework for timeline summarization. *SIGIR 2013 Workshop on Time-aware Information Access (TAIA'2013).*

Van der Lee, C., Krahmer, E., & Wubben, S. (2017). PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences . In *Proceedings of the 10th International Conference on Natural Language Generation* (pp. 95–104). Association for Computational Linguistics.

Vaudry, P.-L., & Lapalme, G. (2013). Adapting SimpleNLG for Bilingual English-French Realisation. In *Proceedings of the 14th European Workshop on Natural Language Generation* (pp. 183–187). Association for Computational Linguistics.

Vicente, M., Barros, C., & Lloret, E. (2017). A Study on Flexibility in Natural Language Generation Through a Statistical Approach to Story Generation. In F. Frasincar, A. Ittoo, L. M. Nguyen, & E. Métais (Eds.), *Natural Language Processing and Information Systems. NLDB 2017* (pp. 492–498). Cham: Springer International Publishing.

Vicente, M. E., Barros, C., Agulló, F., Peregrino, F. S., & Lloret, E. (2015). La generacion de lenguaje natural: análisis del estado actual. *Computación y Sistemas, 19*(4).

Vicente, M. E., Barros, C., & Lloret, E. (2018). Statistical language modelling for automatic story generation. *Journal of Intelligent and Fuzzy Systems, 34*(5), 3069–3079.

Viethen, J., & Dale, R. (2007). Evaluation in natural language generation: Lessons from referring expression generation. *Traitement Automatique des Langues*, *48*(1), 141–160.

Vodolazova, T., Lloret, E., Muñoz, R., & Palomar, M. (2013). The role of statistical and semantic features in single-document extractive summarization. *Artificial Intelligence Research*, *2*(3), 35–44.

Walker, M. (2007). Share and Share Alike: Resources for Language Generation. In *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation* (pp. 28–30).

Walker, M., Stent, A., Mairesse, F., & Prasad, R. (2007). Individual and Domain Adaptation in Sentence Planning for Dialogue. *Journal Artificial Intelligence Research*, *30*(1), 413–456.

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*(10), 914–952.

Wanner, L., Vrochidis, S., Rospocher, M., Moßgraber, J., Bosch, H., Karppinen, A., Myllynen, M., Tonelli, S., Bouayad-Agha, N., Casamayor, G., Ertl, T., Hilbring, D., Johansson, L., Karatzas, K., Kompatsiaris, I., Koskentalo, T., Mille, S., Moumtzidou, A., Pianta, E., Serafini, L., & Tarvainen, V. (2012). Personalized Environmental Service Orchestration for Quality of Life Improvement. In L. Iliadis, I. Maglogiannis, H. Papadopoulos, K. Karatzas, & S. Sioutas (Eds.), *Artificial Intelligence Applications and Innovations* (Vol. 382, p. 351-360). Springer Berlin Heidelberg.

White, M., A. J. Clark, R., & D. Moore, J. (2010). Generating Tailored, Comparative Descriptions with Contextually Appropriate Intonation. *Computational Linguistics*, *36*(2), 159–201.

Williams, S., & Reiter, E. (2008). Generating basic skills reports for low-skilled readers. *Natural Language Engineering*, *14*(04), 495–525.

Xing, Y., & Fernández, R. (2018). Automatic Evaluation of Neural Personality-based Chatbots. In *Proceedings of the 11th International Conference on Natural Language Generation* (pp. 189–194). Association for Computational Linguistics.

Yan, R., Wan, X., Otterbacher, J., Kong, L., Li, X., & Zhang, Y. (2011). Evolutionary timeline summarization: A balanced optimization framework via iterative substitution. In *In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'11* (pp. 745–754). ACM.

Yu, J., Reiter, E., Hunter, J., & Mellish, C. (2007). Choosing the content of textual summaries of large time-series data sets. *Natural Language Engineering*, *13*(01), 25–49.

Zajic, D., Dorr, B. J., & Schwartz, R. (2004). BBN/UMD at DUC-2004: Topiary. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Workshop on Document Understanding* (pp. 112–119). Association for Computational Linguistics.

Zhang, H., Xu, H., Bai, S., Wang, B., & Cheng, X. (2004). Experiments in TREC 2004 Novelty Track at CAS-ICT. In *Proceedings of the 13th Text Retrieval Conference (TREC).*

Zock, M., & Lapalme, G. (2010). A Generic Tool for Creating and using Multilingual Phrasebooks. In *Natural Language Processing and Cognitive Science, Proceedings of the 7th International Workshop on Natural Language Processing and Cognitive Science, NLPCS 2010, In conjunction with ICEIS 2010, Funchal, Madeira, Portugal, June 2010* (pp. 79–89).