Computing and Informatics, Vol. 36, 2017, 1–31, V 2017-May-15

RECENT ADVANCES IN NATURAL LANGUAGE GENERATION: A SURVEY AND CLASSIFICATION OF THE EMPIRICAL LITERATURE*

Rivindu Perera, Parma Nand

Natural Langauge Processing Group
Centre for Artificial Intelligence Research
Auckland University of Technology
Auckland, New Zealand
e-mail: {rivindu.perera, parma.nand}@aut.ac.nz

Abstract. Natural Language Generation (NLG) is defined as the systematic approach for producing human understandable natural language text based on non-textual data or from meaning representations. This is a significant area which empowers human-computer interaction. It has also given rise to a variety of theoretical as well as empirical approaches. This paper intends to provide a detailed overview and a classification of the state-of-the-art approaches in Natural Language Generation. The paper explores NLG architectures and tasks classed under document planning, micro-planning and surface realization modules. Additionally, this paper also identifies the gaps existing in the NLG research which require further work in order to make NLG a widely usable technology.

Keywords: Natural Language Processing, Document planning, Micro-planning, Surface realization

Mathematics Subject Classification 2010: 68T50, 03B65

* THIS IS THE AUTHOR'S DRAFT VERSION OF THE ARTICLE WHICH HAS BEEN PUBLISHED IN FINAL FORM AT: HTTP://WWW.CAI.SK/OJS/INDEX.PHP/CAI/ARTICLE/VIEW/2017_1_1

1 INTRODUCTION

The rapidly increasing need for human interaction with technology has formed the need for machines to be able to generate language rather work only on understanding natural language that humans have uttered. This necessity is evident in many diverse areas regardless of the domain that they belong to. To address this demand for natural looking machine generated text, the discipline of Natural Language Generation (NLG) was born. NLG was first considered as a subfield of Natural Language Processing (NLP), however it was later turned into a major research area and a discipline of its own.

This paper presents a survey of 12 years of research in NLG, covering the recent significant developments in the area, trends and de facto standards used by different NLG systems. We used International Natural Language Generation conference (INLG), European Workshop on Natural Language Generation (ENLG), Natural Language Engineering journal and Computational Linguistics journal as main sources for the survey.

While early systems follow similar architectural flow during the language generation, the newly developed systems employ various approaches and hybrid models not adhering to a single flow of execution. Therefore, it is worthwhile to investigate these novel paradigms as well. Our goal in this survey is to capture these novel methodologies, trends and models used in NLG as well as to identify some of the hurdles that still remain in the area.

The Section 2 of this survey presents different architectures used to develop NLG systems. We present detailed analysis of major architectural models that have both empirical and theoretical foundations. Section 3 explores the content selection employed in NLG. Section 4 describes document planning, which is one of the significant and an initial step in NLG. Section 5 to Section 7, discusses microplanning tasks in NLG systems analyzing different NLG systems and approaches. Surface realization in NLG is described in Section 8. Finally, we draw the conclusions in Section 9.

2 ARCHITECTURE ANALYSIS

High level architecture plays a crucial role in designing NLG systems as in the case of other types of software development. For NLG this is even more valid as it is based on different forms of data and communication channels. Current systems follow diverse set of architectural models based on empirical and theoretical foundations based on both early and recent researches. These models are categorized into four different clusters. This classification does not imply that current systems strictly adhere to these, but they utilize these architectures as their basic foundations and introduce new elements where applicable.

2.1 Pipeline architecture

Pipeline architecture [67] is based on sequential information flow through 3 major components as depicted in Fig. 1. Pipeline architecture is considered to have roots on early developed sequential architectures described by Horacek [39], McKeown [57] and Abb et al. [1].

As most of the earlier developed NLG systems adhered to this, it is considered that pipeline architecture is the consensus architecture that can be easily and effectively utilized for language generation tasks [73]. There are quite significant

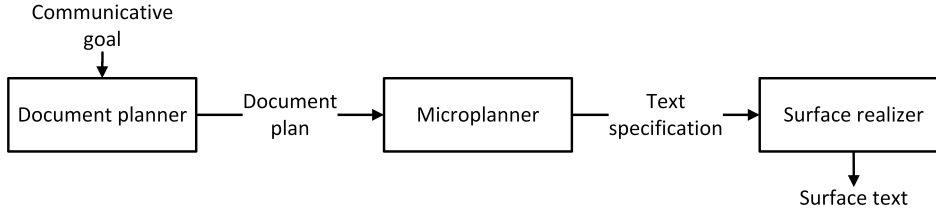


Fig. 1. Pipeline Architecture for NLG

tasks performed within each of the component in pipeline architecture shown in Fig. 1. Table 1 shows the list of these tasks categorized under three main modules in pipeline architecture. The following is a description of the tasks.

Module	Content Task	Structure Task
Document planning	Content determination	Document structuring
Microplanning	Lexicalization	Aggregation
	Referring expression generation	
Realization	Linguistic realization	Structure realization

Table 1. List of tasks categorized under three modules in pipeline architecture

- *Content Determination* is responsible for selecting information needed to be communicated through generated text
- *Document Structuring* manages the structure of the information selected from content determination
- *Lexicalization* operates on what words, terms and concepts need to be included in the text
- *Referring Expression Generation* is the process of determining the way that entities must be referred within generated text

- *Aggregation* operation can be executed to structure and order the sentence structures to build a meaningful sentences
- *Linguistic and structure Realization* is accountable for producing final surface text and presenting it based on the requirements. In this discussion we will refer to this as surface realization which includes both aspects of realization.

The majority of systems we analyzed adhere to the pipeline architecture or hold only minor modification which deviates them from original specification.

CORAL system [26] is one that is strictly adheres to pipeline architecture. It is designed to provide natural and fluent navigational assistance by applying NLG techniques in a route finding system. CORAL is composed of document planner, micro-planner and a surface realizer according to a pipeline architecture. The document planner which is context dependent has special structure based on the route finding domain. CORAL requires a route plan which is generated by the route finding algorithm. Responsibility assigned to the document planner is to generate a structured message which can be fed to the micro-planner. This message is represented based on three key elements provided by route finding algorithm - points, directions and paths. Up to this level, CORAL shows custom implementation which is domain specific, the latter steps are significantly aligned with the pipeline architecture.

Further, real world application of pipeline architecture is demonstrated in SUMTIME - MOUSAM [76, 66] weather forecast generator which generates textual description of weather based on data acquired through a prediction system. There are two significant points that is worth mentioning. Firstly, SUMTIME MOUSAM feed time series numerical data to document planner unit. Secondly, and interestingly the realization engine used here does not output surface text as expected by the pipeline architecture. Instead SUMTIME - MOUSAM is configured to generate output in weather sub-language.

According to Reiter et al. [66], SUMTIME - MOUSAM described above is one subsystem of suite of NLG based SUMTIME-products that they have developed. There are altogether three subsystems which represent the SUMTIME suite, SUMTIME - TURBINE [88], SUMTIME - NEONATE [75] and SUMTIME - MOUSAM. As reported in literature, two other subsystems are also developed following pipeline architecture as in SUMTIME - MOUSAM.

However, sequential flow that pipeline architecture follows is widely used due to its simplicity in assigning responsibilities to different components and defining the processing tasks for information [73]. Though this simplicity in pipeline architecture is considered as an advantage in developing large scale NLG applications, there are known disadvantages as well. Firstly, as there is only one-way information flow, output generated by particular component is never revised or refined. Secondly, less communication between different components can produce text with poor quality [73, 58, 53]. For instance, in pipeline architecture there is no communication between content determination unit and lexicalization. The existence of such communication can be easily used to generate better lexicalized text during the process as it has

the knowledge about content that is already chosen. Nevertheless, each component in pipeline architecture needs to generate an intermediate representation in order to communicate with the next. Therefore, this effort has also been taken into discussion as a key disadvantage that pipeline has due to its sequential flow of information processing.

2.2 Revision architectures

Revision architecture model has been proposed in order to overcome the one way interaction flow in pipeline architectures. The model of a revision was first brought into discussion by Vaughan and McDonald [79], but more practical definition can be found in [45]. Later the concept was successfully implemented in NLG systems in early developments [69, 22]. However, its usage in current systems is in a combination with other architectural models. Currently, pure revision based NLG systems are rarely developed.

Revision based development can be employed in NLG systems in two different ways as shown in Fig. 2. In first approach, revision is simply a recursion of the process where each component is invoked with new form of output text. This is hard to implement as it needs complex information processing. Second approach is to associate a separate revision module for each component, so that revision is carried out immediately in the same component. Due to simplicity, second approach is widely used in different NLG systems as well as in combination with other architectures. Recent usage of the revision approach can be found in work carried out

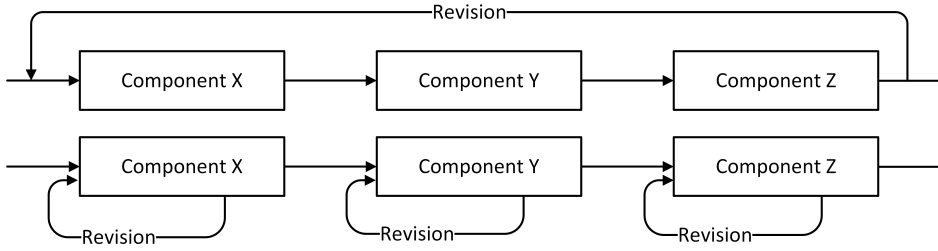


Fig. 2. Two different revision approaches for NLG

by Evans et al. [32]. This novel approach based on text revision uses an unordered tree (termed as generation tree) to revise text, utilizing a tree traversal algorithm. A model of this nature opens ways to investigate text revision in new dimensions rather depending on conventional revision modules. As this work is in progress, Evans et al. [32] have not given experimental results that they have acquired, but model seems to be promising as a revision approach.

Inui et al. [44] discuss text revision as a post-processing task where a shallow paraphrasing technique is employed. In this attempt, researchers employ a paraphrasing engine capable of revising lexical structures. This work is significant in

two ways. Firstly, it shows implementation of a revision module for lexical structures and next the readability ranking model that they present as a classification problem.

There are several other revision based approaches described in NLG [2] which show similar properties and features to those discussed above. However, significant factor noticed in both revision and pipeline architectures, is the way decisions are taken in components. Each component is responsible for a one task, but no collective effort is taken towards the final text generation. This brings the new issue of distributed reasoning where inability to define single reasoning engine for the whole text generation process.

2.3 Uniform architectures

A uniform architecture specifies single reasoning engine to the NLG flow which can solve the issue of distributed reasoning in flow based NLG architectures. Early development of this nature can be found in Knowledge and Modalities Planner (KAMP) system [3] which introduced the concept of multi-agent planning in NLG. GLINDA [47] can also be considered as system which follows the uniform generation process like KAMP. Unlike KAMP, GLINDA is based on set of well defined rule set which are invoked based on defined constraints which brings novelty to the uniform generation process.

Recently uniform generation methodologies which can be considered in this context have also become prominent. Harbusch and Woch [36] present uniform processing architecture for NLG employing Tree Adjoining Grammar. Though single reasoning feature is not specifically represented in this research, still it shows the uniformity through integration which needs to be addressed. Nevertheless, recently proposed NLG architectures consisting of reasoning element borrowed from early developed uniform architectures and mingled with novel models. This can be considered as a positive trend towards NLG architectures.

2.4 Adherence to design principles

Employing software design principles is a current trend in NLG architectures. The usage of software architectural models in NLG has been found to be beneficial as such models are evolving through addressing theoretical and practical issues.

RAGS [58] architecture proposed by Mellish and his team is impressive piece of work performed in this context. In RAGS, researchers emphasize the importance of reusable data resources and formal definition of modules within the NLG architecture. When observing closely, it can be noticed that these concepts are derived from well established software design approaches such as resource re-usability [56] and formal definitions for software components [33]. Two significant elements in RAGS are data definitions and different interfacing operators that RAGS has presented to the NLG research community. It is clear that Reiter's pipeline architecture [67] does not provide formal data model that NLG systems can employ. Instead it is

equipped with a representation of data objects [58]. This gap is very well addressed in RAGS with a well specified data model. The pipeline architecture does not consider interfacing between different modules as a separate process. This makes it difficult to reuse modules in another system. RAGS has achieved this by considering the interfacing as a new operator which can be formally defined. Theoretical definition of RAGS is brought into practical system development through RICHES [18]. RICHES attempts to bring the RAGS goals into wider discussion by showcasing model based and interface based novelties that was presented by RAGS initial theoretical approach.

Compared to RAGS, a more improved applicability of the reusing concept can be found in recent work carried out by Macedo [53] through an implementation of a model driven approach for NLG. Macedo brings three important software design principles to the NLG domain through his model driven approach:

- *consideration of models as reusable assets*
- *separation of business domain concerns and implementation platform concerns*
- *automation of the engineering process*

Out of the three concepts discussed, first one is already considered in RAGS as a primary goal. However, the second and the third aspects hold novelty that NLG architecture development is looking for.

3 CONTENT DETERMINATION

Selecting the content to be delivered plays a crucial role in NLG process as it has a direct impact on the final surface text generated. According to Reiter and Dale [67], there are four aspects that need to be focused on during content determination:

- *selecting data based on significance*
- *summarizing data, so that important information is always included*
- *include information derived through inference*
- *customizing data based on the end-user needs*

Early traditional approaches towards determining the content relied on knowledge bases built specifically for the domain considered [40]. This has then transformed into ontology driven approaches which is widely employed by several early researchers. In most current approaches, content determination is carried out by various methods where some of them have roots on early traditional approaches.

3.1 Machine learning and pattern recognition for content determination

As in many NLP sub-domains, arrival of machine learning and pattern recognition in NLG brought new dimensions to the field.

Duboue and McKeown [31] present the statistical approach implemented towards acquiring content selection rules. This involves the biographical description generation using newly introduced content selection method which automatically derives content selection rules. The core of the research is the measurement of the variation of a language model generated through the clustering. This measurement is used to identify the correlation between word choice and variation of data. The value of this measurement signals whether to include the particular data set being considered as the selected content or not. The methodology presented in this research is more effective than simply employing a knowledge base as employed in traditional approaches. This is because of two reasons. Firstly, statistical analysis is sensitive to data and its variation compared to rule based selection from a knowledge base. Next, this method can be easily generalized to different domains compared to knowledge based approach. Nevertheless, these advantages can be thought of as general advantage that one should expect from a statistical approach towards content selection.

Several similar approaches compared to one described above can be observed in past research attempts. All of them come with different flavours of statistical analysis which makes them unique. In aforementioned Duboue and McKeown's [31] approach, the clustered data are subjected to the correlation operations which actually measures the applicability of the content. Moving few steps further, Barzilay and Lapata [7] present a collective approach towards the content selection where parallel consideration of all entities is performed. It is clear that in this method content classification is deliberated as a problem of collective classification which provides a broader view of the data instead of deep analysis accomplished by Duboue and McKeown [31]. To evaluate the suitability of the model, collection of football data is considered targeting on generating a summary. Providing a comparison with a standard classifier, Barzilay and Lapata [7] argue that collective classification strategy can outperform.

Collective classification introduced in previous research is only considering particular level of grouping. Nevertheless, granularity of this grouping can also affect the content being selected. Empirical analysis of effect on the content selection made by granularity of grouping, is discussed by Kelly et al. [49], which is an extension to the previously discussed model presented by Barzilay and Lapata [7].

As opposed to considering content selection as a classification problem, there is an opportunity to view this as a pattern mining issue. Portet et al. [64] show empirical formulation of pattern mining which is developed as a part of BT-45, a system that provides summaries based on a neonatal intensive care unit. An interesting point that they exaggerate during pattern mining is identification of both short and long term patterns. Resolution of the pattern identification has appeared as a great challenge during this process, but this has been achieved through ignorance or interpolation. Interpretation of these identified patterns and events is performed by associating an abstraction and domain knowledge where inter-linking of events is also made possible. Nevertheless, not enough information present in this research to signal the reader up to what granularity the interpretation is carried out. However,

more detailed description on data interpretation is available in BT-Nurse system [42] which is related to BT-45 system. In BT-Nurse, researchers explain the way that medical knowledge encoded in an ontology can leverage the interpretation of data with a predetermined rule set.

3.2 Rule based and heuristic search for content determination

Employing rules in content selection is also a common approach. Though BT-Nurse (see Section 3.1) utilizes rules partially in its process mixed up with a traditional ontology approach, there exist systems which tend to employ rule based approaches and search mechanisms as the only content selection methodology.

Bouayad-Agha et al. [15] empirically showcase the usage of a rule based methodology through an implementation of a content selection strategy to generate football summaries. In this proposed model, content selection is accomplished in a more granularized approach. The most important factor to notice in this context is rules that they have employed. All rules defined are manually coded based on the determined relevance criteria. Even though, main selection relies on a rule set which is predetermined, relevant measurement during selection is developed with a statistical approach, but it does not significantly contribute toward content selection.

3.3 Employing semantic web for content determination

Recently, semantic web related technologies have grown exponentially and have employed in many different domains. The content selection has also benefited from this new trend. Several researchers have employed semantic web for the content selection task, sometimes mixed with earlier discussed approaches.

First content selection task employing semantic web data was organized by Bouayad-Agha et al. [16] which set up the common ground for other researchers. Task was more clearly focused on Resource Description Framework (RDF) triples which is used extensively by semantic web. It is required by competitors of this shared task to build a working system which can select more relevant triples from given set of triples.

Kutlak et al. [51] present an approach based on a simple heuristic inspired by the common ground principle. In this attempt they assent that more mentioned facts are going to be the best candidates that need to be selected during the content selection. However, implementation based on this heuristic and using FreeBase [13] data has not shown considerable accuracy. Still, this attempt is important as it moves in a different path compared to earlier considered statistical and rule based approaches. Participating in the content selection task, Venigalla and Eugenio [80] bring out another similar approach to the one previously mentioned. In this attempt, the researchers more specifically operate on different predicates which are derived from FreeBase. Clustering applied for predicates is a novel paradigm introduced for the semantic web based content selection. In-house testing of this method shows that it can perform well with basic facts, but poor in other domains.

Application of Linked Data based content selection in open domain environments can be seen in work presented by Perera and Nand [62, 63]. In this approach Perera and Nand [62] employ triple weighting based methodology to rank DBpedia triples and then a threshold based selection to retrieve the finalized selection. Compared to the aforementioned models this approach provides the domain independence for the content selection by reducing the effort on building rules for content selection.

4 DOCUMENT STRUCTURING

Document structuring is the process of turning the selected content to a more structured format, so that information can be passed to next levels for further processing. There are few key properties that document structuring module should satisfy:

- *group* messages, so that it can boost surface text organization
- *order* messages to maintain the communication flow within surface text
- *relate* messages or groups of messages for better linking of surface text

Nevertheless, given these properties of a document structuring module, the way that it attempts to accomplish can vary based on the domain that is considered. For instance, document structuring of semantic representation and structuring applied in a system which handles numerical time series data, will have major differences.

Schemas and rhetorical relations are two different structuring techniques that most of the NLG systems employ. They have their roots on early developed representation models for knowledge which make them suitable for system wide information representation.

4.1 Document structuring using schemas

Structuring mechanism of schemas has its inspiration from early developed knowledge representation schemas such as frames [60] and scripts [71]. When dedicating schemas for special purpose document structuring module, it has evolved as a more productive representation strategy.

Most of the early developed systems utilized Attribute Value Matrices (AVMs) for document structuring. Simple lower level AVM that can be expected from Weather Reporter [67] is shown in Fig. 3. The STOP, another early taken attempt towards generating natural language letters for smokers is also based on AVM styled document structuring.

However, researchers have also come up with other flexible structures which still based on initial concept introduced using attributes and values.

Somayajulu et al. [76] discuss the document structuring unit MOUSAM (a weather forecast generator) which based on tuples of values. Based on five attributes: time, wind speed lower range, wind speed upper range, wind direction and modifiers. Resulting tuple contains only values of above mentioned attributes, for

$$\begin{bmatrix} \text{type: DocumentPlan} \\ \text{children:} \begin{bmatrix} \text{type: ConstituentSet} \\ \text{relation: narrativeSequence} \\ \text{constituent:} \begin{bmatrix} \text{type: RainEventMessage} \\ \text{period:} \begin{bmatrix} \text{month: 04} \\ \text{year: 2014} \end{bmatrix} \\ \text{day:} \begin{bmatrix} \text{day: 16} \\ \text{month: 04} \\ \text{year: 2014} \end{bmatrix} \\ \text{rainType: heavy} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Fig. 3. Attribute Value Matrix (AVM) of a simple weather report

instance $\langle 0700, 8, 13, North, nil \rangle$ can be considered which represent respective values. Though, tuples can work well in numerical information representation, when attempting to represent relations they become inefficient.

Library

$$stock : Copy \mapsto Book$$
$$borrowed : Copy \mapsto Person$$
$$shelved : PCopy$$

$$shelved \bigcup borrowed = domstock$$
$$shevled \bigcap domborrowed = \emptyset$$
$$\forall p : Person.\#(borrowed \rhd p) \leq maxLoan$$

Fig. 4. Z test case for a library specification

Furthermore, it is common that NLG systems processing numerical data and semantic representations find it hard to come up with document structuring. However, systems that are already based on more structured data sources can define messages with less effort. Cristiá and Plüss [24] propose a NLG system that generate natural language descriptions based on Z-test cases. As Z is a more formalized language with strict representation hierarchy, defining a document structure has turned easier than those considered earlier. For instance, considering the example Z test case given in Fig. 4, generation of document structure can be achieved by converting the same representation to map with a schema. Main reason behind this simplicity of conversion is that data itself represent a schema. Nonetheless, this is not a general case for most of the NLG systems.

4.2 Document structuring using rhetorical structures

Rhetorical structure is a result of study performed towards functional approach for text generation introduced by Mann and Thompson [54] where they proposed the Rhetorical Structure Theory (RST). Fig. 5 depicts the basic structure of a RST relation with *nucleus* (central segment of text) and *satellite* (a peripheral segment). Widely used RST relation types in NLG are listed in Table 2 with their definitions. Other than this basic structure there exist multi-nuclear relations in RST which have no central segment of text. For instance, *contrast* is a relation in this nature which is rarely used in NLG.

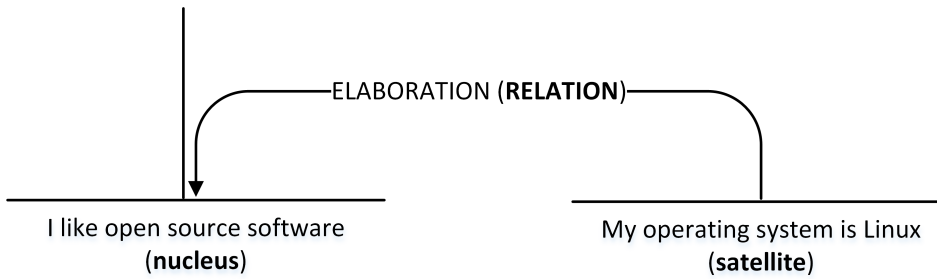


Fig. 5. Basic structure of rhetorical relation

Rocchi and Zancanaro [70] utilize rhetorical structures for their video documentary generation with the limited set of rhetorical relations including *Elaboration*, *Background*, *Sequence* and *Circumstance*. These rhetorical structures are applied on the passages that are used for the document generation. The core technology description that employed for rhetorical relation generation phase which should elaborate on mapping the text to relation, is missing in this research.

However, offering more deep analysis on rhetorical relation mapping process, Theune et al. [77] present their digital storytelling system, Narrator. Narrator seem to be a system that intends to continue the recent trend appeared in the NLG field for producing narratives which is pioneered by systems like ProtoPropp [35], PRINCE [38] and StoryBook [19]. Rhetorical relation types used in Narrator includes *Cause*, *Contrast*, *Temporal*, *Purpose* and *Elaboration*. Key reason that Narrator achieves considerable performance with rhetorical relations is that due to its input (knowledge source), Fabula (a story representation in the form of a causal network) [23]. Fig. 6 depicts the graph view of a Fabula. Fabula itself represents more similarity with RST. This has been a special advantage when employing RST for systems which has knowledge source in the form of event based structures like Fabula.

Relation	Nucleus	Satellite
Background	Text whose understanding is being facilitated	Text for facilitating understanding
Condition	Action or situation whose occurrence results from the occurrence of the conditioning situation	Conditioning situation
Elaboration	Basic information	Additional information
Enablement	An action	Information intended to aid the reader in performing an action
Evidence	a Claim	Information intended to increase the reader's belief in the claim
Circumstance	Text expressing the events or ideas occurring in the interpretive context	An interpretive context of situation or time
Interpretation	A situation	An interpretation of the situation
Justify	Text	Information supporting the writer's right to express the text

Table 2. Commonly used rhetorical relations with definitions [55]

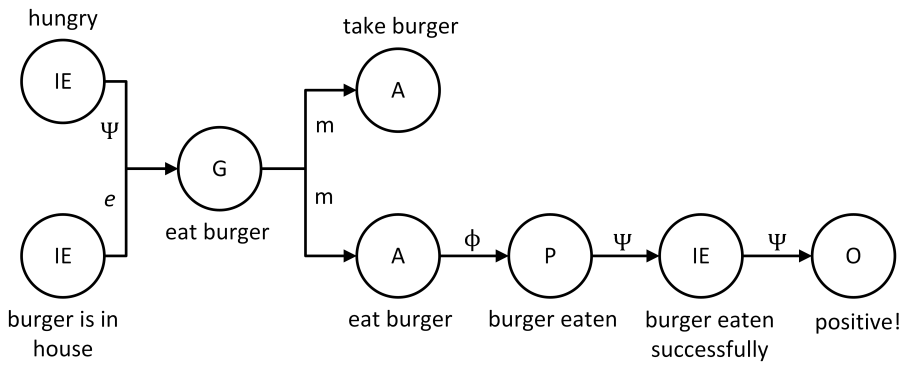


Fig. 6. Graph view of Fabula representation

5 LEXICALIZATION

Lexicalization converts document plan into linguistic structure. This step is critically important in the NLG process as it directly contributes for the human friendliness of the final surface text. Lexicalization can be carried out in two different approaches:

- *coarse grained lexicalization*
- *fine grained lexicalization*

In a coarse grained lexicalization, main focus is placed on generating the lexicalized text in the simplest way possible. Usually, this type of a lexicalization is achieved through a template filling mechanism which ultimately produces a lexicalized structure, but granularity cannot be guaranteed.

Instead of considering lexicalization as a one process, fine grained lexicalization attempts to consider more in depth analysis. It is generally considered that fine grained lexicalization can generate more lexicalized text compared to coarse grained approaches. This is due to significant approaches that are built around fine grained systems such as:

- collocation operations - which checks which words are mostly used in pairs
- redundancy operations - which reduce text by checking unnecessary data
- word choice operations - which choose words that can maximize the effectiveness of communicative goal

5.1 Rule based lexicalization

Due to simplicity, rule based approaches are more common in lexicalization. However, the process of usage of this rule based approach varies from system to system. Danlos et al. [27] demonstrate a more practical approach employed in lexicalization integrated into their EasyText NLG system. EasyText consumes a lexical database built using human involvement, specifically linguists. This fact sounds well in the lexicalization as it is more focused towards real world lexicalization. Conversely, this process is more resource expensive and need special concentration on domain being considered. Once implemented, this model can be easily consumed via simple morphological operations.

Though human involvement in lexicalization is noticed as a distinctive factor, it has its own disadvantages. Humans have different lexical preferences. With limited human involvement, lexicalization can be biased towards lexical preferences of those who involved in evaluation. This arises the issue which stops generalizing the lexicalization of the system even within same domain being considered. Addressing this gap, Reiter et al. [66] present an interesting model which relies on consistent set of data-to-word rules. This involves converting a set of time phrases to linguistic equivalents through a fixed rule and a comparison of these with expert suggested and corpus derived phrases (e.g., 12:00 $\Rightarrow$ by midday, 15:00 $\Rightarrow$ by mid afternoon).

Several other similar approaches for lexicalization are associated with rule based strategies. Specifically, work reported by Siddharthan [72] and Williams and Reiter [87] encompass the usage of choice rules in lexicalization process.

5.2 Lexicalization using case based reasoning

Case Based Reasoning (CBR) has drawn major attention in application where new solutions can be derived from past successful solutions applied. CBR works in four major steps which make them more suitable for a lexicalization process. Below we have explained the process of how lexicalization can benefit from CBR four steps process.

- *Retrieve* past lexicalizations applied. If this is an empty set, then there are options to search in a lexical database to mine similar choices
- *Reuse* the found lexicalizations (if any) in the new scenario by mapping new scenario to the past scenario processed
- *Revise* the new lexicalization by evaluating it using provided metrics
- *Retain* the new lexicalization as a new solution, so that it can be retrieved and reused

Empirical research performed by Hervás and Gervás [37] is recognized as the initial attempt that introduced the CBR to the lexicalization process through a Case Retrieval Net (CRN). In the proposed CRN approach, they particularly focus on a fairy tale generation system which ultimately consumes the novel CRN model developed. Model itself is based on well defined heuristic which wrapped around the foundation of CRN.

5.3 Corpus based lexicalization

Corpus based approaches are relatively common in most of the NLP sub domains. When considering NLG, corpus based approach like collocation (analyzing what words are frequently used as pairs or in combination) is a familiar model. This is mainly because of extensively developed resources that supports corpus based methods. Current corpus based approaches follow more advanced or mix methods for lexicalization than simple collocations.

In seminal work presented by Bangalore and Rambow [6], the authors highlight the importance of sense tagged corpus which can provide lexeme to meaning mappings. Due to absence of this resource they have moved further with an informal approach. However, the issue brought into discussion can be extensively important when working with corpus based approaches. More profound practical work in corpus based lexicalization is presented by Barzilay and Lee [9]. They present multi-parallel corpora which consisted of multiple verbalizations for related semantics. Employing a corpora in this nature can directly benefit the lexical choice operators by opening a rich vocabulary.

5.4 Statistical approaches for lexicalization

Applying statistical methods in lexicalization has also been more frequently reported with a diversity of usage instances. Walker et al. [82] introduce three criteria for more human friendly lexical substitution - *recognizability*, *suitability* and *ambiguity*. Among these three criteria, *recognizability* heavily relies on statistical measurement. As they introduced, *recognizability* attempts to measure the likelihood of a word to appear in the lexicon of the user. This model sounds interesting and effective when compared to earlier discussed collocation model - which does not consider reader's lexicon. Measuring *recognizability* is achieved through Zipf distribution, model which is based on distribution of word occurrence probabilities.

PuppyIR project [11, 5] is also powered by a statistical lexicalization strategy. In this research, Latent Words Language Model (LWLM) is employed obtaining the benefit of better learning of unseen text. However, LWLM role in this research is contributed only partially to the final text being produced, as it also engage the WordNet [59] for synonym generation.

Furthermore, several of other statistical approaches target on generating near-synonyms for lexical substitution. Among them Point-wise Mutual Information (PMI) based approach presented by Inkpen [43] and Latent Semantic Analysis (LSA) based method proposed by Wang and Hirst [84] bring new dimensions to the lexicalizations. Furthermore, unlike the first, latter mentioned LSA approach is a combination with lexical co-occurrence which makes it accurate with a level (74.5%) which moves beyond the baseline accuracy level.

5.5 Ontology based lexicalization

Ontology based approaches are also commonly used in recently developed NLG systems for lexicalization. There are direct benefits which can be gained from an involvement of an ontology during lexicalization:

- *adaptability* : finding an ontology is easier than searching a better corpus for a domain being considered
- *coverage*: unlike a corpus, ontology provides a broader coverage of semantic representation

Cimiano et al. [20] introduce a lexicalization implementation based on ontology lexicon. This new method utilizes a conditional approach which intends to provide the lexical choice with required granularity. According to their example, *"to cut"* can be lexicalized to *"to chop"* if granularity is high, to *"to thinly slice"* if granularity is fine and further. Therefore, they exaggerate that applying suitable conditions is equally important when employing an ontology for lexicalization task. Furthermore, solution for the problem which arises if multiple conditions are satisfied is also provided in this research which is a corpus based collocation operation.

6 REFERRING EXPRESSION GENERATION

Referring an entity in the lexicalized text is also important to produce a smooth flow in surface text. Referring Expression Generation (REG) has two main areas to be considered,

- referring previously mentioned entity and
- generating distinguishing identifier for an entity.

In first case we consider an example text such as *"Barak Obama is the president of United States. He is a graduate of Columbia University"*. We have used personal pronoun *"He"* to refer *"Barak Obama"* where we retained from mentioning the full name again. Further, it is not always the case to use a pronoun and instead referring a shorter version such as "Obama" is also considered as a referring expression. This is important to reduce unnecessary repetition of information which ultimately increases the naturalness of the text.

Consider an instance such as *"switch on the white IPhone 4S"*. If there are multiple *"IPhones"* available to operate, tokens like *white* and *4S*, help the user to uniquely identify which one should be turned on.

Both these scenarios need to be handled when developing NLG systems which require several entities to be mentioned and referred in surface text generated.

6.1 Incremental algorithm and its extensions for referring expression generation

Reiter's initial work [67] in REG has gifted the incremental algorithm to NLG community. Algorithm 1 depicts the original definition of incremental algorithm. L is a set of properties that need to uniquely distinguish the object, P is the complete list of properties to identify objects and C is the list of distractors which need to be eliminated to uniquely identify the object we need.

Consider the example for incremental algorithm given below, where we need to distinguish Phone_1(*White IPhone 4s*) from Phone_2 (*Samsung S3*) and Phone_3 (*Black IPhone 4s*).

1. L is initialized with an empty set
2. C is initialized with Phone_1, Phone_2, Phone_3.
3. Iterate though attribute in P
   (a) Add attribute brand as a modifier ⟨brand, IPhone⟩ (Samsung S3 is removed)
   (b) Add attribute colour as a modifier ⟨colour, White⟩ (Black IPhone 4s is removed)
   (c) At this point C is empty as there are no distractors, and L has a list of properties (e.g.,⟨*brand, IPhone*⟩, ⟨*colour, White*⟩) to distinguish *White IPhone 4s*.

Algorithm 1: Incremental algorithm

$C = \{\langle all\ distractors \rangle\}$
$P = \{\langle the\ list\ of\ useful\ properties \rangle\}$
$L = \{\}$

MakeReferringExpression(C,P)
$L \leftarrow \{\}$
while p_i *of list* P **do**
  if $RulesOut(p_i \neq nill)$ **then**
    $L \leftarrow L \cup p_i$;
    $C \leftarrow C - RulesOut(p_i)$;
  else
    ;
  if $C = \{\}$ **then**
    return L
  else
return failure

RulesOut(p)
$return x : x \in C \wedge \neg hasProperty(x, p)$

Besides systems that employ this algorithm, extended approaches are also common in REG development. Deemter [28] proposed boolean extension for incremental algorithm in which he pointed out inability of backtracking in incremental algorithm. Simply, incremental algorithm cannot add property that is eliminated if it is found that removed one in the most distinguishing one at the end. Furthermore, Deemeter theoretically proves through an example that preference in property selection in REG is subjective in many scenarios. His methodology is based on boolean operations which gradually end up with the most important property list rather performing a simple deduction.

Aforementioned idea is further elaborated through a more empirical approach reported by Deemter et al. [29]. This research dives deep into broad evaluation of incremental algorithm with several combinations of properties. Among the evaluations carried out by Deemter et al. [29], test setting using Furniture sub corpus attempt to provide a clear idea about preference in property selection for incremental algorithm.

Another extension of incremental algorithm is introduced by Kelleher and Kruijff [48] where algorithm is used to generate locative expressions. In this attempt of research, more focus is placed on the context model which signals whether the generated referring expression accurately distinguishes the element need to be referred. Further, they assent that integration of conditions which can increase the relative importance of an object can generate more distinguishing expressions. This model

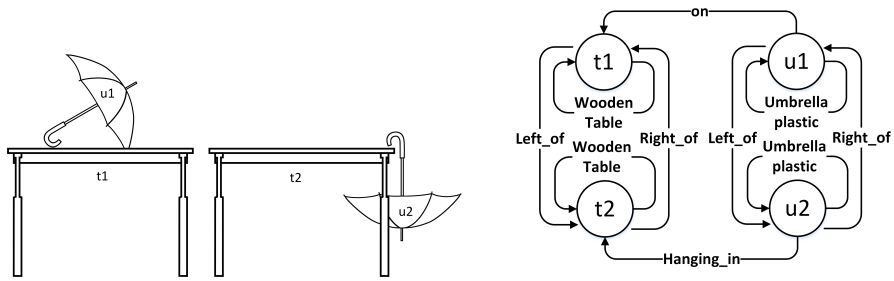
in a way shows some similarities to the earlier mentioned approach presented by Deemter et al. [29].

6.2 Graph based approaches for referring expression generation

Krahmer et al. [50] put forward their initial theoretical foundation of employing labelled directed graph for REG. Method presented by Krahmer et al. [50] is more focused on relational descriptions in REG, where how a particular object can be distinguished in relation to another object appeared in the context. Offering a broad overview of existing approaches, they point out two issues noticed,

- *infinite recursions* - where properties to distinguish targeted object are added recursively generating longer descriptions
- *forced incrementality* - when one relation is failed to distinguish the object, rest are incrementally attempted

However, among these two issues, latter is accepted as a solution in some scenarios, but never the first.



(a) Example scenario (b) Generated graph

Fig. 7. Example scenario and respected graph generated using REG method described by Krahmer et al. [50]

We consider an example scenario based on the Krahmer's theoretical foundation of graph based solution. Fig. 7(a) shows two umbrellas , one on the table and another hanging in the table. Graph generated based on Krahmer's approach is depicted in Fig. 7(b). This allows us to apply various graph based operations to generate the best referring expression for represented object in relation to other objects present. Recommended operation as reported is finding the cheapest (lowest cost) sub-graph which distinguish the target object.

However, Krahmer's initial theoretical foundation does not address the issue when multiple sub-graphs exist in the same cost (algorithm is trained to select the first by default). Recently carried out empirical testing of Krahmer's algorithm reported by Viethen et al. [81] encompass the aforementioned issue and build an

extension which successfully solves the issue with the use of expansion of sub-graphs to be checked.

Based on initial work carried out by Krahmer, different types of graph based solutions are introduced. A more complex generalization is reported by Deemter and Krahmer [30] where Krahmer's initial algorithm is extended to support diverse REG scenarios by target set partitioning.

6.3 Knowledge representation based approaches for referring expression generation

REG has also empirically tested with various knowledge representation models. Croitoru and Deemter [25] introduce Conceptual Graph (CG) based model for REG. Researchers argue that REG can be significantly benefited by employing a CG model because it has a high reputation for its strength in knowledge representation and inference based on different case relations [74].

Usage of Description Logic (DL) can be seen in work carried out by Areces et al. [4] where they attempt to define existing algorithm with a unified DL based approach. Though this research opens ways to rethink REG with a novel perspective, generalizability of the approach is still problematic and stays as an open research question.

Ren et al. [68] carry the DL based approach one step ahead with Ontology Web Language which eventually provides a more extensibility. In this piece of work, open research question left by previous research is attempted using logic based utilities such as relational specification and quantification.

7 AGGREGATION

Aggregation is primarily responsible for generating more structured and grouped sentences. Among other NLG components, aggregation is a relatively less focused component. Yet it contributes for the quality of the final surface text significantly. There are several occasions where aggregation is performed as a sub module in realization, but still it has the ability to appear as individual module and interact with realization later in the process.

Depending on the level of aggregation required for a NLG task, aggregation can range from lowest to the highest complexity. However, it is also believed that complex and accurate aggregation can increase the usability of the final surface text produced.

Reiter and Dale [67] encompass four types of aggregation operations which range from lowest complexity to the highest complexity. We demonstrate the difference and applicability of these operations considering four sentences ($S_1 - S_4$) below.

S_1: Peter is a boy.
S_2: Peter is a high school student.

S_3: Peter is a good chess player.
S_4: He is a good cricket player.

- *simple conjunction*, where multiple linguistic elements are connected using a connective
  (e.g., Peter is a high school student and he is a good cricket player)
- *shared participant*, where shared object is used for combination
  (e.g., Peter is a high school student and a good cricket player)
- *shared structure*, where shared structure is used for combination
  (e.g., Peter is a good chess and cricket player)
- *syntactic embedding*, where embedding relevant information is performed generating complete structure
  (e.g., Peter is a good chess and cricket playing high school student)

Moving beyond these operations, some researches employ various computational models in aggregation. This section place special focus on such paradigms that bring novelty to the domain.

7.1 Trainable approaches for aggregation

Walker et al. [83] present a model for aggregation which is based on repeated generation of text. Ranking function is employed to decide whether additional repetitions need to be carried out before the final text output. Model is also evaluated with the use of a travel system [65] which showcases different aspects of the novel model.

Though this sort of a strategy sounds well for the aggregation, it suffers from several severe issues. In first place, this approach does not consider how applicable the text is for the end user. Walker and his team encompass that human involvement can be associated in ranking process, but the involvement cannot always represent the choice of different users. This is moreover a common issue which can be noticed in many other models. Furthermore, according to researchers, eight different clause combination operations are provided for the aggregation process. These are based on standard four types elaborated above and some new ones introduced. However, the problem is these rules do not provide the coverage for most tricky scenarios. Specially, considering the adjective based syntactic embedding is defined only for a limited domain and no further improvement is noticed.

There is also an opportunity to view aggregation as an issue which needs a supervised approach. Barzilay and Lapata [8] formulate the hypothesis that targeted on finding a cluster of phrases from the given set which maximizes the defined utility function. This approach has many similar properties as one that is noticed earlier. Instead of a utilizing a ranking function for the regeneration like in the earlier model, in this research utility function is employed. Main difference lies in the way that text is fed, Barzilay and Lapata [8] use clustered text and Walker et al. [83] use raw text.

7.2 Evolutionary algorithms for aggregation

Evolutionary algorithms have started playing a key role in any aggregation where fine optimization is generally required. Recently evolution strategy based aggregation attempts have moved beyond baseline expectations.

Hervás and Gervás [37] present evolution approach based on three operations, crossover, mutation and aggregation. Among these, in this context we focus on aggregation operator which primarily developed towards clause aggregation. The aggregation specified by this is more biased towards the conjunction and disjunction process based on the structure reformulation. However, there is always a chance for regeneration based on their fitness functions incorporated. Three of the fitness functions out of five defined are targeted towards the aggregation - redundant attributes, coherence and overlooked information function. However, the most important factor that they exemplify here, is that the number of generations. They have found that nearly 50 generations are needed for a better aggregated and accurate description.

Nevertheless, similar approach to one above is reported by Hua and Mellish [41], where attempt is made towards more coherent description. In terms of the process and evaluation, genetic algorithm brought by this has considerable similarity with model described by Hervás and Gervás [37]. However, the feature selection has attempted to make significant difference with previous evolution strategy.

Hua and Mellish [41] employ semantic relations, shifting operations and more importantly embedding operations which derived from early attempts in aggregation.

7.3 Graph based aggregation

Graph based solutions are common in NLG systems. An attempt to perform aggregation as a hyper-graph is reported by Bayyarapu [10]. Involvement of hyper-graph is clearly justified by the researchers stating that its applicability to perform conjunctive operations more easily compared to other models such as Conceptual Graphs (CGs) which need complex projection operations. Example hyper-graph is depicted in Fig. 8 based on the three sentences mentioned in the start of this section.

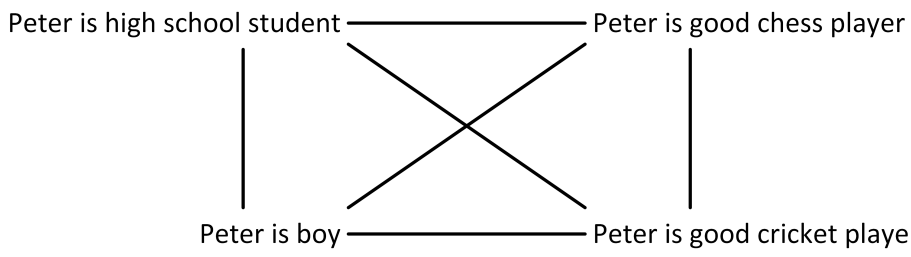


Fig. 8. Example hyper-graph representation

Once the hyper-graph is built, the task of aggregation is a graph traversal problem and finding the best subgraph which can represent the required aggregation. This can be achieved in different traversal algorithms available, but Bayyarapu [10] involves a context sensitive discriminative model which can provide the aggregation probability. This approach is a novel hybrid model that is presented by this research for the aggregation which stands few steps ahead of other attempts we discussed.

8 REALIZATION

Realization or more commonly referred to as surface realization is the task of mapping the text specification to the surface form sentences which will be ultimately presented to readers. Realization is sometimes considered as two steps: structure and linguistic realization. In this section we consider realization as whole covering both aspects of realization.

Most of the early NLG tasks such as, displaying an error message, displaying the full name after getting first name and last name from the user and other preconfigured text displaying system were equipped with the most basic realization unit. This simplest realization is often called as canned text which still being used in different applications which do not employ complex NLG tasks.

Alternatively, there is another approach which can be used as filling mechanism. This sort of a filling mechanism is based on templates which involved structures predetermined based on the program requirements. Template mechanisms are also still in use, but NLG has moved beyond these simple strategies of text realization approaches.

Currently, realization is driven by two camps, statistical and grammar. Among these two, grammar based approaches have drawn considerable attention recently. This section analyses both these approaches in terms of their recent advances and modifications which bring novelty to the NLG domain.

8.1 Statistical approaches for realization

As seen in previously analyzed components, statistical approaches in realization is also playing a major role. Statistical techniques for realization are often mingled with grammar based approaches. Therefore, some of the approaches which we present here still are backed by grammar based specifications. However, in next section we totally focus on pure grammar based realization models.

Langkilde-Geary [52] exemplifies HALogen system which relies highly on statistical model, specifically a n-gram language model. HALogen is also powered by a symbolic generator, but in this context we will mainly focus on statistical model HALogen has introduced. Corpus based language model introduced in HALogen is basically targeting the best generated text. Nevertheless, several similar approaches can be noticed in work reported by Clallaway [21], Nakanishi et al. [61] and Cahill and Genabith [17].

Belz et al. [12] provide a more detailed and broad view of statistical realization attempts and report that such techniques have made reusable realization frameworks. This is true for many other components as well, when statistical techniques are incorporated, but their applicability need to be assessed in a more effective way focusing on linguistic structures of language.

8.2 Grammar based realization

Pure grammar based approaches are common in realization. This is primarily due to ease of implementation based on the grammar specification unlike statistical approaches which need rigorous evaluation to generate the best. In the context of realization, systemic grammar and functional unification grammar are used effectively in several research attempts.

Systemic grammar specification [46] is one that can be easily employed for a language generation task. Provided such specification, surface text realization can be carried out using conjunctive operations following the rules defined. However, challenge that need to be achieved while using a such grammar specification is that specifying lexicons. Most of the grammar based systems specify own lexicons as a dataset and then consult it during generation. For instance, SimpleNLG [34] use a lexicon set of several records and consume it during generation to decide words to be used with surface text. SimpleNLG is currently utilized for German [14] and used in several NLG implementations [78, 42].

OpenCCG [85, 86] is another grammar based realizer. OpenCCG developers employ Combinatory Categorial Grammar (CCG) with defined combinatory rules. Special feature in OpenCCG is that its categories are organized into lexical families which make it unique from conventional CCG. However, added advantage that OpenCCG developers achieved from this strategy is that ease of providing specifications for lexicon. In realization this advantage has also ultimately contributed for effective processing based on lexicon families.

9 SUMMARY AND CONCLUSION

This paper presented a detailed survey on NLG and its components focus towards generating descriptive text. Our discussion started with analyzing architectures and continued with analyzing diverse attempts carried out to develop NLG tasks. All major NLG tasks were analyzed considering most recent developments and state-of-the-art techniques introduced during last 12 years. During our investigation, it was noticed that unlike early research, current NLG focused research has benefited from variety of models and approaches. This has created a need for a more specialized survey attempt into recent developments. As there exist previous survey attempts, this attempt should be seen as a similar survey for the current period. We have attempted to provide wide and deep coverage of most of the significant developments within NLG arena stating from earliest works up to the most recent developments.

Acknowledgements

This material is based on a study funded by the School of Computer and Mathematical Sciences, Auckland University of Technology to build a Natural Language Generation framework.

REFERENCES

[1] Abb, B.—Gunther, C.—Herweg, M.—Maienborn, C.—Schopp, A.: Incremental Syntactic and Phonological Encoding – An Outline of the Synphonics Formulator. In: Proceedings of the Fourth European Workshop on Natural Language Generation, Pisa, Italy, 1993.

[2] Aluísio, S. M.—Specia, L.—Pardo, T.—Maziero, E.—Fortes, R.: Towards Brazilian Portuguese automatic text simplification systems. In: Proceedings of the ACM symposium on Document engineering, New York, USA, 2008.

[3] Appelt, D.: Planning English Sentences. Cambridge University Press, Cambridge, UK, 1985.

[4] Areces, C.—Koller, A.—Striegnitz, K.: Referring expressions as formulas of Description logic. In: Proceedings of the Fifth International Natural Language Generation Conference, Salt Fork, OH, USA, 2008.

[5] Azzopardi, L.—Glassey, R.—Mounia, L.—Polajnar, T.—Ruthven, I: PuppyIR: Designing an Open Source Framework for Interactive Information Services for Children. In: Proceedings of the Third Annual Workshop on Human-Computer Interaction and Information Retrieval, Washington DC, USA, 2009.

[6] Bangalore, S.—Rambow, O.: Corpus-based Lexical Choice in Natural Language Generation. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, 2000.

[7] Barzilay, R.—Lapata, M.: Collective Content Selection for Concept-to-Text Generation. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Morristown, NJ, USA, 2005, pp. 331–338.

[8] Barzilay, R.—Lapata, M.: Aggregation via set partitioning for natural language generation. In: Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Morristown, NJ, USA, 2006, pp. 359–366.

[9] Barzilay, R.—Lee, L.: Bootstrapping Lexical Choice via Multiple-Sequence Alignment. In: Proceedings of the Empirical methods in Natural Language Processing, Pennsylvania, USA, 2002, pp. 164–171.

[10] Bayyarapu, H.: Efficient Algorithm for Context Sensitive Aggregation in Natural Language Generation. In: Proceedings of the Recent Advances in Natural Language Processing, Hissar, Bulgaria, 2011, pp. 84–89.

[11] Belder, D.—Deschacht, K.—Moens, M.: Lexical Simplification. In: Proceedings of the First International Conference on Interdisciplinary Research on Technology, Education and Communication, Kortrijk, Belgium, 2010.

[12] Belz, A.—White, M.—Genabith, J.—Hogam, D.—Stent, A.: Finding Common Ground: Towards a Surface Realization Shared Task. In: Proceedings of the Sixth International Natural Language Generation Conference, Dublin, Ireland, 2010.

[13] Bollacker, K.—Evans, C.—Paritosh, P.—Sturge, T.—Taylor, J.: Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, New York, USA, 2008.

[14] Bollmann, M.: Freebase: Adapting SimpleNLG to German. In: Proceedings of the Thirteenth European Workshop on Natural Language Generation, Nancy, France, 2011, pp. 133–138.

[15] Bouayad-Agha, N.—Casamayor, G.—Wanner, L.: Content Selection from an Ontology-based Knowledge base for the Generation of Football Summaries. In: Proceedings of the Thirteenth European Workshop on Natural Language Generation, Nancy, France, 2011, pp. 72–81.

[16] Bouayad-Agha, N.—Casamayor, G.—Wanner, L.—Mellish, C.: Overview of the First Content Selection Challenge from Open Semantic Web Data. In: Proceedings of the 14th European Workshop on Natural Language Generation, Sofia, Bulgaria, 2013, pp. 98–102.

[17] Cahill, A.—Genabith, J.: Robust PCFG-Based Generation Using Automatically Acquired LFG Approximations. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, 2006, pp. 1033–1040.

[18] Cahill, L.—Carroll, J.—Evans, R.—Paiva, D.—Power, R.—Scott, D.—Deemter, K.: From RAGS to RICHES: Exploiting the Potential of a Flexible Generation Architecture. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, Toulouse, France, 2001, pp. 106–113.

[19] Callaway, C.—Lester, J.: Narrative prose generation. Artificial Intelligence, Vol. 139, 2002, No. 2, pp. 213–252.

[20] Cimiano, P.—Lüker, J.—Nagel, D.—Unger, C.: Exploiting Ontology Lexica for Generating Natural Language Texts from RDF Data. In: Proceedings of the Fourteenth European Workshop on Natural Language Generation, Sofia, Bulgaria, 2013, pp. 10–19.

[21] Clallaway, C.: Evaluating Coverage for Large Symbolic NLG Grammars. In: Proceedings of the International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003.

[22] Cline, B.—Nutter, J.: Kalos - A System for Natural Language Generation with Revision. In: Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, USA, 1994.

[23] Cobley, P.: Narratology - The Johns Hopkins Guide to Literary Theory and Criticism. Johns Hopkins University Press, Baltimore, USA, 2005.

[24] Cristiá, M.—Plüss, B.: Generating Natural Language Descriptions of Z Test Cases. In: Proceedings of the Sixth International Natural Language Generation Conference, Dublin, Ireland, 2010, pp. 173–177.

[25] Croitoru, M.—Deemter, K.: A Conceptual Graph Approach to the Generation

of Referring Expressions. In: Proceedings of the International Joint Conference on Artificial Intelligence, Hyderabad, India, 2007, pp. 2456–2461.

[26] Dale, R.—Geldof, S.—Prost, J.: CORAL: Using Natural Language Generation for Navigational Assistance. In: Proceedings of the 26th Australasian computer science conference, Darlinghurst, Australia, 2003, pp. 35–44.

[27] Danlos, L.—Meunier, F.—Combet, V.: EasyText: An Operational NLG System. In: Proceedings of the 13th European Workshop on Natural Language Generation, Nancy, France, 2011, pp. 139–144.

[28] Deemter, K.: Generating Referring Expressions: Boolean Extensions of the Incremental Algorithm. Computational Linguistics, Vol. 28, 2002, No. 1, pp. 37–52.

[29] Deemter, K.—Gatt, A.—Sluis, I.—Power, R.: Generation of Referring Expressions: Assessing the Incremental Algorithm. Cognitive Science, Vol. 36, 2012, No. 5, pp. 799–836.

[30] Deemter, K.—Krahmer, E.: Graphs and Booleans: On the Generation of Referring Expressions. Computing Meaning. Vol. 83, 2006, No. 1, pp. 17–53.

[31] Duboue, P.—McKeown, K.: Statistical Acquisition of Content Selection Rules for Natural Language Generation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Morristown, USA, 2003, pp. 121–128.

[32] Evans, R.—Weir, D.—Carroll, J.—Paiva, D.—Belz, A.: Modelling Control in Generation. In: Proceedings of the Eleventh European Workshop on Natural Language Generation, Saarbrücken, Germany, 2007, pp. 25–32.

[33] Garlan, D.: Formal Modeling and Analysis of Software Architecture: Components, Connectors, and Events. In: Proceedings of the Formal Methods for Software Architectures, Bertinoro, Italy, 2003, pp. 1–24.

[34] Gatt, A.—Reiter, E.: SimpleNLG: A Realisation Engine for Practical Applications. In: Proceedings of the Twelfth European Workshop on Natural Language Generation, Athens, Greece, 2009, pp. 90–93.

[35] Gervás, P.—Díaz-Agudo, B.—Peinado, F.—Hervás, R.: Story plot generation based on CBR Knowledge-Based Systems. Knowledge-Based Systems, Vol. 18, 2005, No. 4, pp. 235–242.

[36] Harbusch, K.—Woch, J.: Integrated Natural Language Generation with Schema-Tree Adjoining Grammars. In: Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Mexico, 2002.

[37] Hervás, R.—Gervás, P.: Case Retrieval Nets for Heuristic Lexicalization in Natural Language Generation. In: Proceedings of the Twelfth Portuguese Conference on Artificial Intelligence, Covilhã, Portugal, 2005, pp. 55–66.

[38] Hervás, R.—Pereira, F.—Gervás, P.—Cardoso, A.: Cross-domain Analogy in Automated Text Generation. In: Proceedings of the Third Joint Workshop on Computational Creativity, Riva del Garda, Italy, 2006.

[39] Horacek, H.: The Architecture of a Generation Component in a Complete Natural Language System. Academic Press, London, UK, 1990.

[40] Hovy, E.: Automated Discourse Generation Using Discourse Structure Relations. Artificial Intelligence, Vol. 63, 1993, No. 2, pp. 341–385.

[41] Hua, C.—Mellish, C.: Capturing the Interaction between Aggregation and Text Planning in Two Generation Systems. In: Proceedings of the First International Conference on Natural Language Generation, Mitzpe Ramon, Israel, 2000.

[42] Hunter, J.—Freer, Y.—Gatt, A.—Reiter, E.—Sripada, S.—Sykes, C.: Automatic Generation of Natural Language Nursing Shift Summaries in Neonatal Intensive Care: BT-Nurse. Artificial Intelligence in Medicine, Vol. 56, 2012, No. 3, pp. 157–172.

[43] Inkpen, D.: A Statistical Model for Near-synonym Choice. ACM Transactions on Speech and Language Processing, Vol. 4, 2007, No. 1, pp. 1–17.

[44] Inui, K.—Fujita, A.—Takahashi, T.—Iida, R.—Iwakura, T.: Text Simplification for Reading Assistance: A Project Note. In: Proceedings of the Second International Workshop on Paraphrasing, Sapporo, Japan, 2003.

[45] Inui, K.—Tokunaga, T.—Tanaka, H.: Text Revision: A Model and its Implementation. In: Proceedings of the Sixth International Workshop on Natural Language Generation: Aspects of Automated Natural Language Generation, Trento, Italy, 1992.

[46] Jurafsky, D.—Martin, J.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, Upper Saddle River, NJ, USA, 2000.

[47] Kantrowitz, M.—Bates, J.: Integrated Natural Language Generation Systems. In: Proceedings of the Sixth International Natural Language Generation Conference: Aspects of Automated Natural Language Generation, Trento, Italy, 1992.

[48] Kelleher, J.—Kruijff, G.: Incremental Generation of Spatial Referring Expressions in Situated Dialog. In: Proceedings of the Twenty-First Joint International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney Australia, 2006.

[49] Kelly, C.—Copestake, A.—Karamanis, N.: Investigating Content Selection for Language Generation using Machine Learning. In: Proceedings of the Twelfth European Workshop on Natural Language Generation, Athens, Greece, 2009, pp. 130-137.

[50] Krahmer, E.—Erk, S.—Verlag, A.: Graph-Based Generation of Referring Expressions. Computational Linguistics, Vol. 29, 2003, No. 1, pp. 53–72.

[51] Kutlak, R.—Mellish, C.—Deemter, K.: Content Selection Challenge - University of Aberdeen Entry. In: Proceedings of the Fourteenth European Workshop on Natural Language Generation, Sofia, Bulgaria, 2013, pp. 208–209.

[52] Langkilde-Geary, I.: An Empirical Verification of Coverage and Correctness for a General-Purpose Sentence Generator. In: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, Philadelphia, USA, 2002.

[53] Macedo, H.: Model Driven Development Approach to Natural Language Generation Systems. ACM SIGSOFT Software Engineering Notes, Vol. 35, 2010, No. 1, pp. 1–7.

[54] Mann, W.—Thompson, S.: Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. Text, Vol. 8, 1988, No. 3, pp. 243–281.

[55] Mann, W.—Taboada, M.: Introduction to Rhetorical Structure Theory. Simon Fraser University, Upper Saddle River, Burnaby, Canada, 2005.

[56] McArthur, K.—Saiedian, H.—Zand, M.: An Evaluation of the Impact of Component-based Architectures on Software Reusability. Information and Software Technology, Vol. 44, 2002, No. 6, pp. 351–359.

[57] McKeown, K.: Text Generation using Discourse Strategies and Focus Constraints to Generate Natural Language Text. Cambridge University Press, Cambridge, UK, 1985.

[58] Mellish, C.—Scott, D.—Cahill, L.—Paiva, D.—Evans, R.—Reape, M.: A Reference Architecture for Natural Language Generation Systems. Natural Language Engineering, Vol. 12, 2006, No. 1, pp. 1–34.

[59] Miller, A.: WordNet: A Lexical Database for English. Communications of the ACM, Vol. 38, 1995, No. 11, pp. 39–41.

[60] Minsky, M.: A Framework for Representing Knowledge. Massachusetts Institute of Technology, Massachusetts, USA, 1974.

[61] Nakanishi, H.—Miyao, Y.—Tsujii, J.: Probabilistic Models for Disambiguation of an HPSG-Based Chart Generator. In: Proceedings of the Ninth International Workshop on Parsing Technology, Vancouver, British Columbia, 2005, pp. 93–102.

[62] Perera, R.—Nand, P.: Real Text-CS-Corpus Based Domain Independent Content Selection Model. In: Proceedings of the IEEE 26th International Conference on Tools with Artificial Intelligence, Limassol, Cyprus, 2014, pp. 599–606.

[63] Perera, R.—Nand, P.: The Role of Linked Data in Content Selection. In: Proceedings of the 13th Pacific Rim International Conference on Artificial Intelligence, Gold Coast, Australia, 2014, pp. 573–586.

[64] Portet, F.—Reiter, E.—Gatt, A.—Hunter, J.—Sripada, S.—Freer, Y.—Sykes, C.: Automatic Generation of Textual Summaries from Neonatal Intensive Care Data. Artificial Intelligence, Vol. 173, 1995, No. 8, pp. 789–816.

[65] Rambow, O.—Rogati, M.—Walker, M.: Evaluating a Trainable Sentence Planner for a Spoken Dialogue Travel System. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 2001, pp. 426–433.

[66] Reiter, E.—Sripada, S.—Hunter, J.—Yu, J.—Davy, I.: Choosing Words in Computer-generated Weather Forecasts. Artificial Intelligence, Vol. 167, 2005, No. 2, pp. 137–169.

[67] Reiter, E.—Dale, R.: Building Natural Language Generation Systems. Cambridge University Press, Cambridge, UK, 2000.

[68] Ren, Y.—Deemter, K.—Pan, J.: Generating Referring Expressions with OWL2. In: Proceedings of the Twenty-Third International Workshop on Description Logics, Waterloo, Canada, 2010.

[69] Robin, J.: A Revision-Based Generation Architecture for Reporting Facts in their Historical Context. In: Proceedings of the New Concepts in Natural Language Generation: Planning, Realization and Systems, London, UK, 1993, pp. 238–265.

[70] Rocchi, C.—Zancanaro, M.: Generation of Video Documentaries from Discourse Structures. In: Proceedings of the Ninth European Workshop on Natural Language Generation, Budapest, Hungary, 2003.

[71] Schank, R.—Abelson, R.: Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures. Psychology Press, London, UK, 1977.

[72] Siddharthan, A.: Complex Lexico-syntactic Reformulation of Sentences Using Typed Dependency Representations. In: Proceedings of the Sixth International Natural Language Generation Conference, Dublin, Ireland, 2010, pp. 125–133.

[73] Smedt, K.—Horacek, H.—Zock, M.: Architectures for Natural Language Generation: Problems and Perspectives. In: Proceedings of the Fourth European Workshop on Natural Language Generation, Pisa, Italy, 1996, pp. 17–46.

[74] Sowa, J.: Conceptual Graphs as a Universal Knowledge Representation. Computers and Mathematics with Applications, Vol. 23, 1992, No. 2, pp. 75–93.

[75] Sripada, S.—Reiter, E.—Hunter, J.—Yu, J.: Summarizing Neonatal Time Series Data. In: Proceedings of the Tenth Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary, 2003, pp. 167–170.

[76] Sripada, S.–Reiter, E.—Davy, I.: SUMTIME-MOUSAM: Configurable Marine Weather Forecast Generator. Expert Update, Vol. 6, 2003, No. 3, pp. 4–10.

[77] Theune, M.—Slabbers, N.—Hielkema, F.: The Narrator: NLG for Digital Storytelling. In: Proceedings of the Eleventh European Workshop on Natural Language Generation, Dagstuhl, Germany, 2007, pp. 109–112.

[78] Thomas, K.—Sripada, S.: Atlas.txt: Linking Geo-referenced Data to Text for NLG. In: Proceedings of the Eleventh European Workshop on Natural Language Generation, Dagstuhl, Germany, 2007, pp. 163–166.

[79] Vaughan, M.—McDonald, D.: A Model of Revision in Natural Language Generation. In: Proceedings of the 24th annual meeting on Association for Computational Linguistics, Morristown, USA, 1986, pp. 90–96.

[80] Venigalla, H.—Eugenio, B.: UIC-CSC: The Content Selection Challenge Entry from the University of Illinois at Chicago. In: Proceedings of the 14th European Workshop on Natural Language Generation, Sofia, Bulgaria, 2013, pp. 210–211.

[81] Viethen, J.—Mitchell, M.—Krahmer, E.: Graphs and Spatial Relations in the Generation of Referring Expressions. In: Proceedings of the Fourteenth European Workshop on Natural Language Generation, Sofia, Bulgaria, 2013, pp. 72–81.

[82] Walker, A.—Siddharthan, A.—Starkey, A.: Investigation into Human Preference between Common and Unambiguous Lexical Substitutions. In: Proceedings of the Thirteenth European Workshop on Natural Language Generation, Dublin, Ireland, 2011, pp. 176–180.

[83] Walker, M.—Rambow, O.—Rogati, M.: SPoT: A Trainable Sentence Planner. In: Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies, Stroudsburg, USA, 2001, pp. 1–8.

[84] Wang, T.—Hirst, G.: SPoT: Near-synonym Lexical Choice in Latent Semantic Space. In: Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, 2010, pp. 1182–1190.

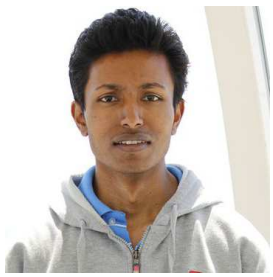
[85] White, M.: CCG Chart Realization from Disjunctive Inputs. In: Proceedings of the Fourth International Natural Language Generation Conference, Sydney, Australia, 2006, pp. 1–8.

[86] White, M.: Efficient Realization of Coordinate Structures in Combinatory Categorial Grammar. Research on Language and Computation, Vol. 4, 2006, No. 1,

pp. 39–75.

[87] Williams, S.—Reiter, E.: Generating Readable Texts for Readers with Low Basic Skills. In: Proceedings of the Tenth European Workshop on Natural Language Generation, Aberdeen, Scotland, 2005, pp. 1–5.

[88] Yu, J.—Reiter, E.—Hunter, J.—Mellish, C.: Choosing the Content of Textual Summaries of Large Time-series Data Sets. Natural Language Engineering, Vol. 13, 2005, No. 1, pp. 25–49.



Rivindu Perera received his B.Eng.(Hons), degree in software engineering from University of Westminster, UK. He has nearly seven years of experience in developing natural language processing applications in various domains. He is now a Ph.D. candidate and a research assistant at the Centre for Artificial Intelligence Research, Auckland University of Technology. His current research project, RealText, is focused on developing a scalable and open domain natural language generation framework. His research interests include natural language processing, text generation, information retrieval, and Semantic Web.

Parma Nand received his Ph.D. in Computer Science from Auckland University of Technology, New Zealand. He is currently a Senior Lecturer and Researcher in the areas of Text Mining and natural Language Processing and leads the Natural Language Processing group in the School of Computer and Mathematical Sciences at AUT. His current research interests include Text Mining and Information Retrieval, particularly in the area of Social Media Text applications. Some of the current projects are online bullying detection, topic tracking in Twittersphere, and Location mining from Tweets.