IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 5 July 2023, accepted 6 August 2023, date of publication 14 August 2023, date of current version 17 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3304903

SURVEY

A Review of Text-to-Animation Systems

NACIR BOUALI[1,3] **AND VIOLETTA CAVALLI-SFORZA**[2]

[1]Data Management and Biometrics, University of Twente, 7522 NB Enschede, The Netherlands
[2]School of Science and Engineering, Al Akhawayn University in Ifrane, 53000 Ifrane, Morocco
[3]School of Computing, University of Eastern Finland, 80101 Joensuu, Finland

Corresponding author: Nacir Bouali (n.bouali@utwente.nl)

ABSTRACT Text-to-graphics systems encompass three types of tools: text-to-picture, text-to-scene and text-to-animation. They are an artificial intelligence application wherein users can create 2D and 3D scenes or animations and recently immersive environments from natural language. These complex tasks require the collaboration of various fields, such as natural language processing, computational linguistics and computer graphics. Text-to-animation systems have received more interest than their counterparts, and have been developed for various domains, including theatrical pre-production, education or training. In this survey we focus on text-to-animation systems, discussing their requirements, challenges and proposing solutions, and investigate the natural language understanding approaches adopted in previous research works to solve the challenge of animation generation. We review text-to-animation systems developed over the period 2001-2021, and investigate their recent trends in order to paint the current landscape of the field.

INDEX TERMS Natural language interface, natural language understanding, computer graphics, semantic parsing, visual semantics.

I. INTRODUCTION

Creating graphical resources is a fastidious and time-consuming task, it requires expertise in computer graphics and programming. Graphical resources are however very useful in various tasks, like advertising, entertainment [1], and education [2]. While discussing the importance of animation in education, Jancheski highlights the difficulty of creating such resources for classroom use, and cites as a challenge that the skills needed to create such resources are beyond a single teacher's domain of knowledge [3]. This makes it desirable to develop a paradigm wherein graphical resources can be created from natural language.

Various research works targeted graphics generation from natural language. They come in two types: text-to-scene and text-to-animation. On the one hand, text-to-scene systems generate 3D scenes from natural language input [1]. The challenges in such systems are related to the spatial relationships between objects, which are usually expressed through prepositions [1]. Text-to-animation systems, referred to hereafter as TTA, on the other hand, are systems that generate 2D or 3D animations from natural language, they extend text-to-scene systems by adding dynamicity to the scenes [4], [5]. The challenge in such systems is related mainly to the visualization of events, which are usually expressed through verbs [4]. In their efforts to map natural language to the semantic representation of animations, TTA systems face many challenges related to natural language input, graphics generation, or to the connection between them [4]. The efforts are however worthwhile, as TTA systems can be very useful in the education and entertainment industry. Education can benefit from TTA systems to create animations to support classroom activities, such as digital storytelling.

Systems that offer the capability of animation generation consist mainly of two separate modules: A Natural Language Understanding NLU module, and a graphics module [6], [7].

It is difficult to categorize TTA systems due to the interdisciplinary nature of the research supporting them. But the underlying generic architecture allows us to characterize these systems as a connection between Natural Language Processing NLP and computer graphics [6], as shown in Figure 1.

In this paper, we aim to investigate TTA systems from a natural language understanding perspective, by systematically

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Wei.

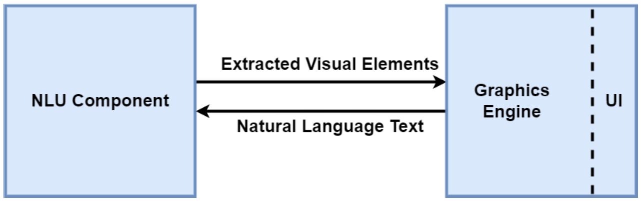


FIGURE 1. A basic architecture of a TTA system.

reviewing TTA research works from the literature published between 2001 and 2021. We define TTA systems as types of systems where the users can describe the environment, the actors, the actions and the props via natural language descriptive prose. Consequently, research works that describe either systems where the user can control an avatar in a virtual environment, as in [8], or systems where the user communicates with a conversational agent as in [9] have been excluded due to the different natural language processing challenges they present.

The rest of this paper is organized as follows. In Section II, we outline the methodology we employed for conducting this review, elaborating on the inclusion and exclusion criteria and explaining our extraction scheme. In Section III, we present the previous research works, attempting to classify text-to-graphics systems and how and where those efforts crosscut with the present paper. Section III continues by discussing the challenges related to the task of TTA, and Section IV leverages these challenges to study the selected works in further depth. We discuss our results in Section V and draw up our conclusions in Section VI.

II. REVIEW STRATEGY

This review adheres to the guidelines established by Kitchenham and Charters [10]. To access the scientific databases and the reviewed material, we relied on the search services of the University of Eastern Finland library's electronic resources.

To begin our review, we examined previous surveys that addressed text-to-graphics systems, namely [6] and [11]. The existing literature recognizes the difference between text-to-scene and text-to-animation tools, yet none of the reviews focuses exclusively on TTA systems. We have, thus, decided to study this category of text-to-graphics systems, with an emphasis on their requirements, challenges while studying and evaluating the effectiveness of the existing solutions.

A. SPECIFYING THE RESEARCH QUESTIONS

By reviewing the existing TTA systems, this review aims to answer the following research questions:

- *What are the requirements of a reliable text-to-animation system?*
- *What challenges have to be solved to enable the animation generation capability?*
- *How do the existing TTA systems approach and fare against the identified challenges?*

To address the research questions, we conducted a comprehensive analysis of the current text-to-animation systems from different perspectives. We first started by identifying the specific requirements and challenges associated with each system, taking into account the diverse domains in which the TTAs are applied. We, subsequently, delved into understanding the processes and pipelines of semantic parsing within the TTA systems, seeking to comprehend the underlying mechanisms and techniques employed in extracting meaning from the input descriptions. Lastly, the study focused on the rendering process, exploring how the generated animation information was communicated and transformed into visually adequate animations through the collaboration between the NLU and the rendering modules.

B. REVIEW PROTOCOL

Kitchenham and Charters suggest that a review protocol is essential to conduct a systematic review [10]. By creating a review protocol in advance, the likelihood of a research bias can be reduced, such as the possibility of the researcher selecting particular studies based on personal biases. We developed a review protocol based on Kitchenham's guidelines. Our protocol included the research questions, inclusion and exclusion criteria, search strategy, data extraction and synthesis methods.

C. SEARCH STRATEGY

We conducted a comprehensive literature search using the following databases: *ACM Digital Library*, *IEEE Xplore Digital Library*, *ScienceDirect*, and *Google Scholar*. We used phrases comprising different combinations of the terms ‘‘animation generation’’, ‘‘natural language visualization’’, ‘‘text-to-animation’’, ‘‘natural language animation’’, ‘‘text-to-graphics’’, ‘‘animation’’, ‘‘language visualization’’, ‘‘graphics generation’’, and ‘‘natural language description visualization’’.

D. STUDY SELECTION

The inclusion criteria were carefully formulated to ensure the relevance of the selected studies. We took into account English-language peer-reviewed journal articles, conference papers, books, and dissertations published between 2001 and 2021. We focused on publications that reported on TTA systems that fulfilled the following requirements:

- *input natural language, and*
- *output 2D, 3D or immersive visualizations, and*
- *have an explicit NLU stage, and*
- *are implemented and not mere designs.*

Figure 2 displays a flowchart depicting the selection criteria process. We conducted a thorough review of all identified publications, including their titles, abstracts, keywords, and contents. We selected articles that appeared relevant to visualizations based on natural language for further analysis. We, however, excluded research on TTA systems based on end-to-end neural networks, including those that involve inputting

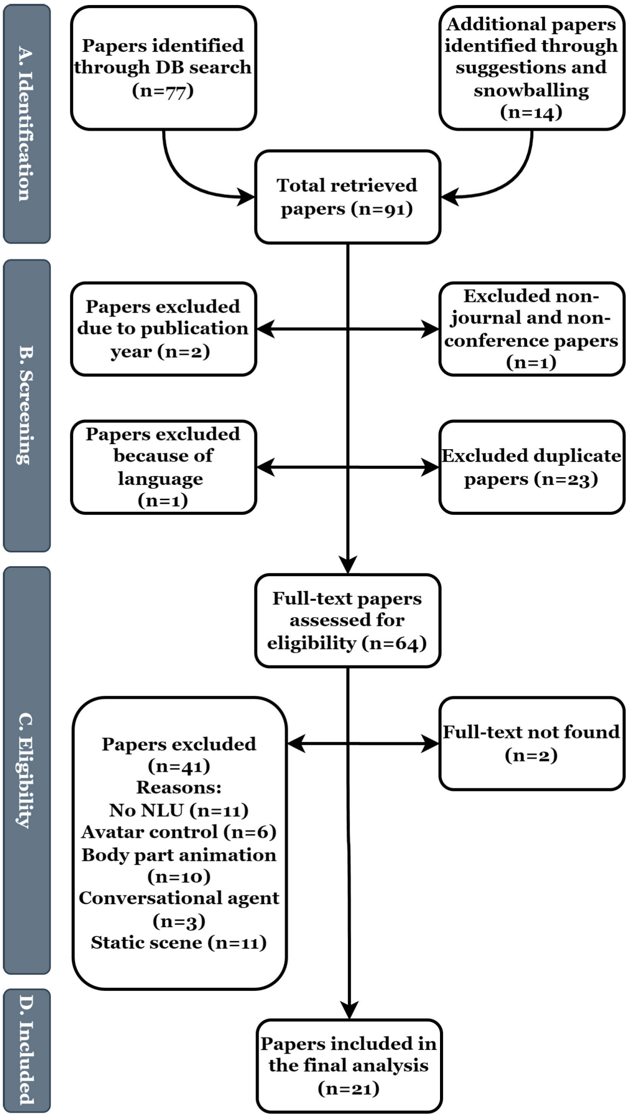

A. Identification
B. Screening
C. Eligibility
D. Included
Papers identified through DB search (n=77)
Additional papers identified through suggestions and snowballing (n=14)
Total retrieved papers (n=91)
Papers excluded due to publication year (n=2)
Excluded non-journal and non-conference papers (n=1)
Papers excluded because of language (n=1)
Excluded duplicate papers (n=23)
Full-text papers assessed for eligibility (n=64)
Papers excluded (n=41)
Reasons:
No NLU (n=11)
Avatar control (n=6)
Body part animation (n=10)
Conversational agent (n=3)
Static scene (n=11)
Full-text not found (n=2)
Papers included in the final analysis (n=21)

FIGURE 2. Flow Diagram of the review process.

raw motion capture (mocap) data (as seen in [12], [13], [14]) or systems that input natural language, such as TGANs-C [15] and CRAFT [16]. This exclusion stems from the absence of an explicit NLU stage in such approaches. Additionally, we excluded articles that concentrated on controlling avatars through speech or text commands. Research on face animation and talking heads was also excluded from this review due to the different NLU challenges it tackles. Initially, 91 research papers were identified through the predefined search keywords on the four databases. However, after applying the inclusion and exclusion criteria, only 21 articles met the selection criteria, as shown in Table 2.

E. DATA EXTRACTION

To facilitate the process of data extraction from the reviewed papers, we established an extraction scheme which focused on three dimensions, human interaction, NLU and rendering.

On the human interaction aspect, we extracted information on the supported input languages, the application domain (education, theatrical preproduction, etc.), and we looked at how the domain influenced the nature of the sentences used (imperative vs. declarative). From the NLU aspect, our extraction focused on the algorithms used to perform the semantic parsing task, which is critical to understand which animation to generate. This has been augmented with an extraction phase of the various knowledge bases that the TTA systems use to augment the input and fill in the blanks the user might leave out while describing an animation. Lastly, in our rendering perspective, we sought to extract the information related to the graphics engines or libraries used to create the animations as well as extract the formal languages the NLU module uses to communicate with the rendering engine.

Before we proceed to the results obtained from these extraction processes in IV, we set out to present the requirements and challenges of animation generation systems in III.

III. REQUIREMENTS AND KEY CHALLENGES IN ANIMATION GENERATION

Text-to-graphics systems have been a hot topic of research recently. The progress made in both the fields of NLP and computer graphics have facilitated the design and development of such technology. Prior to entering into a detailed review of individual research works we set out the requirements that an animation generation system must satisfy and the challenges encountered in building one. We partially base this exposition and the subsequent discussion on existing surveys, adding research that was not previously reviewed and creating a new framework within which to examine prior work.

A. PRIOR AND CURRENT REVIEWS

Hassani et al. reviewed 26 text-to-graphics systems, distinguishing between text to-picture systems, which generate 2D images in response to some natural language query, text-to-scene systems, which generate 3D scenes, and TTA systems, which generate 2D or 3D animations from natural language [6]. The TTA systems were reviewed from a user-interaction and NLP/NLU perspectives [6]. Similarly, Zakraoui et al. reviewed 10 text-to-picture systems from a user-interaction and technical perspectives [11]. They used the term text-to-picture to encompass text-to-picture, text-to-scene and TTA systems.

The current review targets research works that have been published between 2001 and 2021, which means that it overlaps with some of the systems surveyed in both [6] and [11]. As shown in Figure 3, we have surveyed 21 TTA systems (blue), eight of which have already been reviewed in [6] (orange), and one has been reviewed in both [6] and [11](green). In this review, we place an emphasis on the challenges to TTA systems, as well as the requirements and tasks of such systems. Our review of both the challenges and the systems is seen from a storytelling perspective. We have used the identified challenges to rate the TTA systems, and

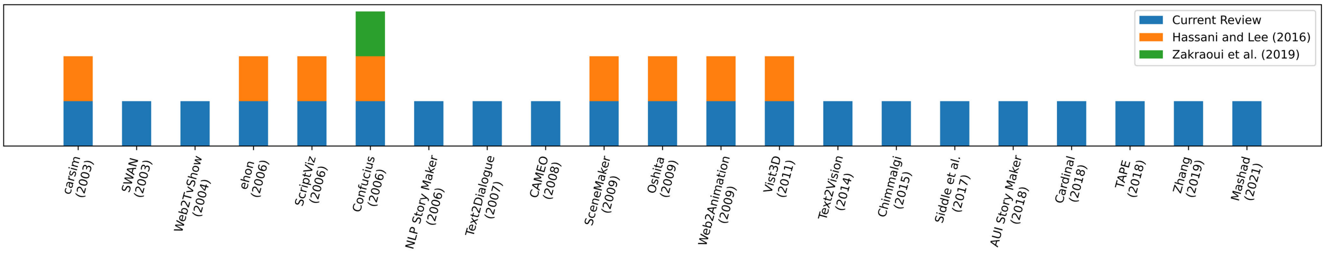


FIGURE 3. Crosscut with previous reviews.

assess how capable the research works examined were of providing reliable text-to-animation.

B. OVERVIEW OF REQUIREMENTS AND COMPONENTS OF TTA SYSTEMS

To generate an animation from natural language text, the system should be able to ground natural language into a formal description that can be mapped to an animation, including all the information required for the animation to be complete. Our research focuses more on verbs, considering that verbs are the essence of event visualization. Hassani et al. identified three requirements for text-to-graphics systems [6]:

- A graphics engine to render the output.
- A natural Language interface able to convert natural language input into a formal description.
- An architecture to put together the two previous requirements of the output.

Hanser et al. identified six tasks for TTA systems [5]:

- Interpret natural language input and extract semantics with a focus on the emotional aspects essential for visualization.
- Integrate a knowledge base for common sense reasoning, affective reasoning and decision rules.
- Map language elements to visual elements.
- Generate the virtual scenes, with 3D scenes and audio and non-audio speech.
- Coordinate the timing of different media.
- Apply cinematography principles to adjust lighting and camera positioning, etc.

The tasks identified above have a specific focus, that is emotions. We, however, believe that the tasks identified are key to TTA systems in general regardless of their domains or foci.

In their efforts to create a TTA system to convert Chinese children stories into computer animations, Lu and Zhang developed SWAN. It relies on a methodology which they refer to as FLICA that divides the process of animation generation into 8 stages, as follows [7]:

- Understanding natural language text and grounding it into a formal semantic representation.
- Doing the story analysis and commonsense reasoning.
- Qualitative planning of display elements: characters, environments, props, etc.
- Director planning.
- Qualitative camera planning.
- Qualitative light and color planning.
- Quantitative camera, light and color planning.
- Cartoon generation based on the quantities identified in the previous step and a knowledge base.

It is worth noting that Lu and Zhang's system, SWAN, allows the user to intervene during the animation generation process [7].

TTA systems (that fulfill the requirements identified by Hassani and Lee, that can provide the tasks as specified by Hanser et al., and that follow the stages identified by Lu and Zhang) face a number of challenges occurring at various stages of the process of animation generation.

We extracted the challenges from the research works surveyed in this article, and provided a classification of these challenges based on the architectural component that tackles them. We classify the challenges as either related to the natural language input, the animation generation or the connection between the two, which we refer to in this article as visual reasoning challenges. Refer to Figure 4 below for a taxonomy of TTA challenges, which are further detailed in the subsections below.

C. CHALLENGES WITH NATURAL LANGUAGE INPUT

Our focus in this section is on NLP issues related solely to TTA systems; consequently, typical NLP issues like co-reference resolution, syntactic parsing [17], etc., are excluded from this analysis. Some issues with the input in TTA systems relate to the domain for which they are developed, and the size of the input provided (single or multiple sentences). These too are not treated.

Furthermore, some of the issues identified at the input level may have repercussions at other levels in the TTA systems architecture, as is the case with "underspecification" which generates an endless list of problems at the level of the visual reasoning entity.

1) IDENTIFICATION OF SENTENCES FOR ANIMATION

In trying to map a sentence to its visual representation (or one of its possible visual representations), an effort is dedicated to converting the input to a formal language that captures the

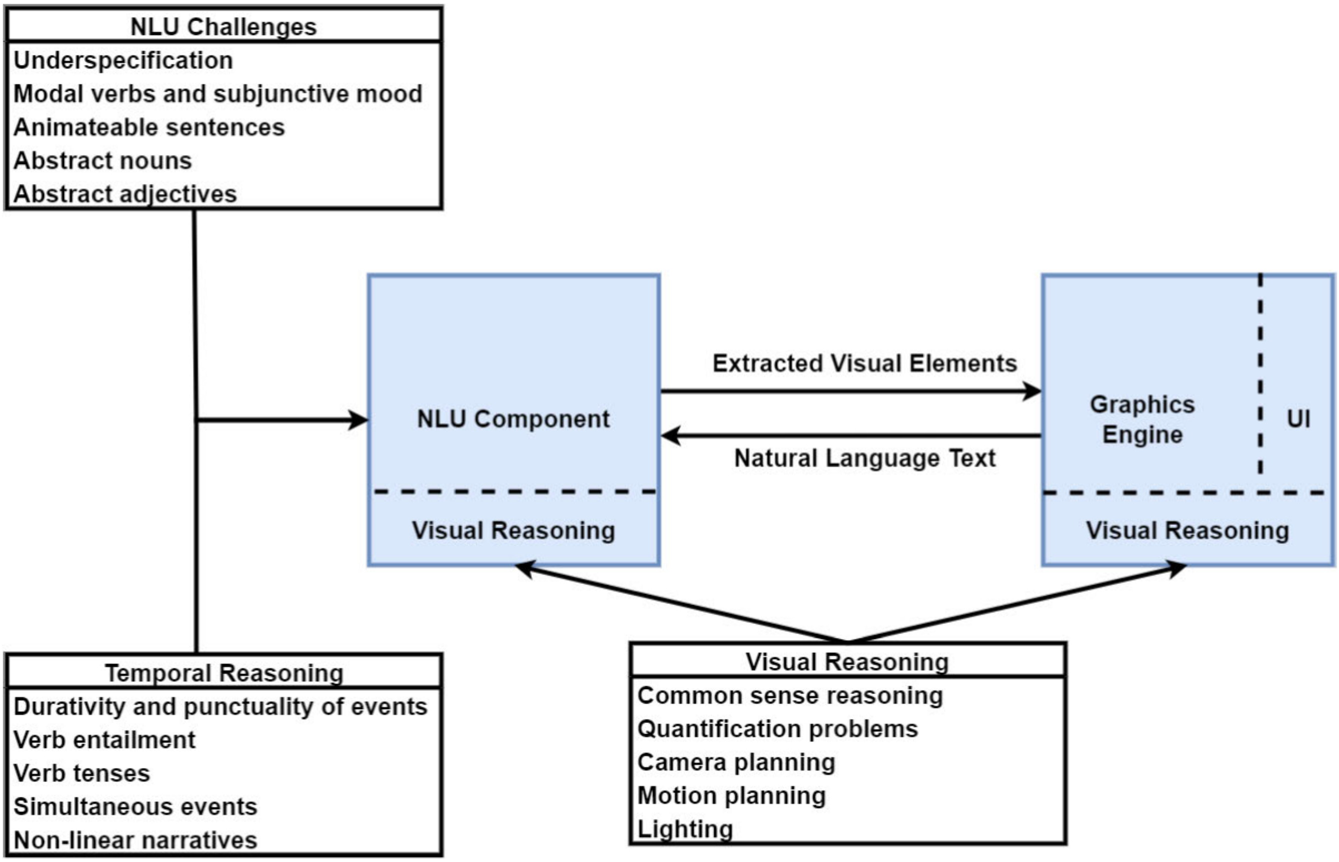


FIGURE 4. TTA challenges by component.

essence of the visual elements it describes. For instance, if the user enters a very forward and simple example such as "John reads a book", a basic NLP module should be capable of generating the following XML representation:

<actor>John</actor>
<action>read</action>
<object>book</object>

In this case, it is possible to map the sentence to a visual representation with an actor reading a book. Other examples present the challenge of verbs describing states, such as "want", "love" or "hate", in which visualization becomes a more difficult task. Consider for example the sentence "John wants a book". Unlike action verbs, such as "read", "walk" or "talk", stative verbs, such as "want", "love" or "hate" denote no visual change in an object's state, the visualization of which might prove difficult or impossible in some cases.

A bigger challenge to animation generation stems from the use of abstract nouns, as in "John wants justice", and figures of speech, like metaphors. When pointing to someone's cowardice, one might use the expression "the man was a chicken". Rendering the sentence literally does not convey the intention. In general, a main issue in animation generation is identifying sentences that cannot be mapped to animations.

2) MODAL VERB AND SUBJUNCTIVE MOOD VISUALIZATION

Modal verbs such as "could", "should", "would", etc. can introduce events in a possible or imaginative world. An animation for "John can read a book" is different from "John reads a book", as the latter describes an ongoing event whilst the former describes a possible event [4]. Similarly, the subjunctive mood which is used to explore hypothetical situations poses a similar challenge. A sentence such as "If John could read a book" should yield a different output than the affirmative statement "John reads a book".

The polysemic nature of modal verbs adds to the complexity of handling this class of verbs in a TTA system. Ruppenhofer and Rehbein worked on annotating the sense of English modal verbs in order to create a dataset for NLP models to be trained on [18]. They distinguish between six senses, shown in Table 1, that the modal verbs can take.

Unlike TTA systems designed for storytelling, whose inputs typically include modal verbs, a TTA system that processes instructional texts (e.g., generating animations from recipes) is less likely to process sentences with modal verbs.

3) NOUN VISUALIZATION

Nouns in sentences can be mapped to 4 different entities, actors, scenes, props or time. While the ability to visualize any of these entities relies mainly on the database of graphical

TABLE 1. Senses of modal verbs.

Sense	Use	Modal	Example
Epistemic	Describes a possible state of the world.	can, may, must, ought, shall	John could have read the book.
Dynamic	Describes the ability of a subject to perform an action.	can	John could have read the book.
Deontic	Gives permission.	can, may	You may leave.
Optative	Expresses a hope or a wish.	May	May the force be with you.
Concessive	Considers an event as a given.	May	Though he may look naive, he's actually very intelligent.
Conditional	Expresses a condition.	Shall	Should you need help, let me know.

Table is based on [18].

resources the TTA uses, one of the challenges is related to a distinction that needs to be made between abstract and concrete objects. Consider as an example the two sentences, "John is approaching" and "the deadline is approaching".

4) ADJECTIVE VISUALIZATION

For visualization tasks, adjectives can be classified in two types: Visually observable or visually unobservable. Visually observable attributes can be object's states or attributes, human attributes such as feelings (happiness, anger, etc.) or others (old, young, etc.). Some visually unobservable attributes can be perceivable by audio means, such as noisy or calm. They can also be perceivable by haptic modalities such as cold, hot, etc. as much as they can be abstract modalities like good, kind or mean, etc. [4]. Some of the latter could be given stereotypical visualizations, as in comic strips, but, in general, the system should be able to distinguish adjectives that can be mapped to an object/actor property and adjectives that cannot be visualized through the animation.

5) UNDERSPECIFIED SENTENCES

Underspecification [4] concerns sentences with correct syntax and semantics but that fail to explicitly communicate every aspect needed for the generation of an animation. Consider the same sentence: "John reads a book". The sentence fails to deliver information on the location of John, the book he's reading, how far is he in the book or even what does John look like [7]. Despite the triviality of the character's appearance, in contextualization efforts–that is when a TTA system is to be used in a specific context (country and target audience)–the looks of John will gain in importance. For example, portraying John as a man from western countries could be accepted if the animation is targeted to a western setting, but using an eastern character (Yemeni for example) might just fall short in meeting the believability criteria of the animation (if any are set). The contrary may be true if the setting of the story is Yemen.

D. TEMPORAL REASONING

1) DURATIVITY AND PUNCTUALITY OF EVENTS

Vendler classifies verbs based on their temporal semantic features. He suggests classes which not only contrast stative verbs to dynamic (action) verbs, but also studies the dynamic verbs and how they behave over the time axis [19]. Vendler describes stative verbs as verbs denoting events with no change. Vendler then classifies dynamic verbs based on their telic or atelic features. He consequently distinguishes between verbs with an endpoint (telic), such as "build a house" or "paint a wall", wherein the action of building ends when the house is built and the action of painting ends when the wall is painted. This kind of actions Vendler contrasts to actions with no end point (atelic), like "walk", "run", or "drive a car", wherein the walking, running or driving actions do not necessarily have an endpoint. Atelic verbs can become telic under specific conditions, for example "walk" is an atelic verb while "walk to" is telic [19]. A problem that TTA systems need to address is distinguishing such temporal semantic features to determine the duration of animations depicted by eventive verbs. In TTA systems which process single sentences at a time, atelic verbs are less of a challenge. The user decides the duration in which the action will be executed, as they can interrupt it by entering a new sentence. The problem becomes however more challenging for TTA systems that process multiple sentences at a time, and so the animator needs to decide how long an atelic verb will be running before the next sentence gets run.

2) VERB ENTAILMENT

An entailed meaning can be inferred from a logical sequence of verbs, in a way that a verb referring to an action cannot be used without the other [20]. For instance, the action of "waking" up entails a prior state of "sleep", "divorce" implies a prior state of "marriage".

3) NON-LINEAR NARRATIVES

When users are allowed to input multiple sentences into a TTA system, a space is created for the challenge of non-linear narratives. That is, a story input by the user may contain events that do not necessarily happen in a sequential order, or that can happen simultaneously. Consider the following input, "Before John read his book, he had to find his glasses." In this effort, a study of the temporal expressions is necessary to be able to infer the order of events over the time axis.

4) VERB TENSES

Tenses of verbs used in a sentence can present a greater challenge in animation generation than they do in text-to-scene or text-to-picture systems, consider the following sentences:
- John reads a book.
- John read a book.
- John is reading a book.
- John has been reading a book.

If these sentences are provided to a TTA system, they would result in the same animation, but problems will arise if any of these sentences is provided in a sequence of events as expected in a multiple-sentence input. Consider the example "John drinks a soda. John bought the drink when he was at the store." The TTA system should be able to reorganize the actions accordingly on the time axis. In the absence of the temporal expressions, the verb tenses give a clear indication on which action precedes which.

5) SIMULTANEOUS ACTIONS

The NLU is not the only challenge to animation generation; even animated sentences present a challenge in visualization. Consider the following sentence: "Sara reads a book while John walks the dog." The NLP module should be able to convert the NL representation of the sentence to its corresponding XML representation for which two animations can be generated for both sentences. The XML representation should also tag that the two animations overlap on the time axis. The display, however, should happen simultaneously, which means that there should be a mechanism to either show both animations in a split screen or focus on the actions of the main character.

E. CHALLENGES IN THE CONNECTION BETWEEN NL AND GRAPHICS: VISUAL REASONING

1) COMMON SENSE REASONING

The key task of the NLU module in a TTA system is grounding natural language into a machine readable representation of the visual elements required to map the NL description to an adequate visualization. TTA systems use various techniques to extract visual elements from texts. Most of the systems reviewed below analyze the parse tree of NL sentences and assign visual semantic roles to each element they are able to render in the animated scene.

Even a successful semantic analysis leaves out some information necessary for the animation. In the SWAN system, Lu and Zhang draw the example of the Snow-White Story, from which they quote "The new queen killed Snow White with a poisonous apple." The NL description, according to the SWAN system, leaves many questions unanswered, for example, where did the killing took place or what was the color of the apple. A common problem with TTA systems, or with text-to-graphics in general doesn't only concern the visual elements the descriptions of which are missing, but involves also visual elements that were not included in the text in the first place, like what was the weather like or was there any sort of dialogue involved between the characters, did the queen convince snow white to eat the apple, etc. [7].

Issues in underspecification, as stated above, go beyond missing information in the text, but crosscut with ambiguity. The example drawn in SWAN raises issue on how the "killing" or "rescuing" actions were performed, as the verbs used to describe these actions are generic. Ma provides a solution for this problem by specifying a Level of Detail (LoD), and distinguishes three levels for an event. A high level to which she refers as the "event level" ("to go" for example), a middle level which is also known as "manner level" ("to walk"), and finally a lower level which identifies the "troponymy level" ("to swagger") [4].

TTA systems that input single sentences add a different challenge compared to systems that take multiple sentences. For instance, a small paragraph like "John saw an animal in the zoo which he didn't recognize. He asked his sister Sara about it and she told him it was a rhino." In visualizing the input sentence by sentence, the graphics engine wouldn't know which animal to render as a response to the first sentence, the identity of the animal remains unknown until the next entry of the text. TTA systems processing multiple sentences are able to infer such information by analyzing the whole text.

SWAN goes further in analyzing the challenges in visual reasoning, by drawing the attention of the TTA researchers to issues related to the lighting in the scene, coloring and camera planning [7].

2) QUANTIFICATION PROBLEM

Ruqian et al. identified a set of issues with TTA systems relating to the quantities of objects in a given scene [21]. The authors examined problems related to quantification while developing the "Shakespeare" system, a cartoonification tool that takes Chinese children stories as input and generates their corresponding animations [21]. In sentences like "There were cars parked on the street", the number of objects is left undetermined in the text, leaving the decision to the common-sense reasoning module. Quantification problems concern also issues like exaggeration as in "There were like a billion people out there", or even accurate statements like "There are 1.5 billion people in China". A system cannot instantiate that many objects in a scene as they wouldn't fit in and it would be a huge load on the device's processing and co-processing units. The Shakespeare system identified further issues with quantification to identify non-countable entities. For instance, sentences like "There were 2 kilograms of apple on the table" present a challenge to the animation process though the weight presents an accurate measure, it cannot be visualized except as an explicit label.

Uncertainty presents another challenge, where according to Ruqian a sentence like "there are two fathers and two sons" can mean that there are 4 characters in the scene, as well as

3 if you consider that there is a father, a grandfather and a grandchild in the scene [21].

3) CAMERA PLANNING

Ruqian and Zhang highlight that viewers can see a plot from different angles and viewpoints. Such a possibility can only be afforded by planning multiple cameras accordingly. In scenes with multiple actors, and props, camera positioning dictates what you see of the scene from where and when [7]. This problem is not trivial and has been tackled elsewhere in the scientific literature, as in [22], [23], [24], and [25].

4) MOTION AND PATH PLANNING

When describing events in an animation, it is often the case that a simple description of a movement does not describe how the movement is carried out in space. Ruqian and Zhang draw the example of describing a cat's movement from the window to the top of a tree. While it might be befitting to simply write *Run(cat, under(window), top(tree))*, such a command however is ineffective when there are obstacles in between the tree and the window [7]. Examples of these are known glitches in video games, wherein players can walk through objects (walls, trees or other players) due to problems in the objects' colliders. Path planning is key to generate natural, believable animations.

IV. ANALYSIS OF THE SYSTEMS

The metrics we decided upon to analyze the identified TTA systems draw from their generic architecture. Every TTA system has a NLU module, which grounds the natural language input into a machine-readable representation, usually a semantic a representation, a task wherein semantic parsing plays a crucial role. TTA systems also rely on a graphics module, which renders the animation, and lastly a visual reasoning module which connects both modules.

To analyze the NLU module, we study how each system performs the semantic parsing task. We start by looking at how the syntactic analysis is performed, and then how each visualization element is captured. Table 3 provides the results of this analysis. Semantic parsing is usually not enough to convert text to animation. We wanted to look at how (parts of) the visual reasoning are carried out in the TTA tools, so we studied the knowledge base (or bases) each system uses to feed in information absent from the text (underspecification) but required for the animation. The results of this analysis are also detailed in Table 3.

The graphics module has been studied from two main perspectives: which markup language is used to identify the visualization elements (background, action, actor, etc.) and which graphics library or graphics engine is used to actually render the animation to the user. We focus on markup languages to identify how the NLU passes animation elements to the graphics engine, and to study whether the markup language is sufficiently human-readable to allow the user to modify the output as they see fit before the animation is rendered. The results of this analysis are captured in Table 4.

To the graphics, NLU and visual reasoning aspects, we have added a fourth dimension: user interaction. In this dimension, we studied how users interact with each system, in which language, for what purpose (domain) and using which type of sentences. Figures 5-7 illustrate the user interaction aspect of this study. While Table 2 identifies the systems by name, and provides a description of each system very briefly.

A. OVERVIEW OF TTA SYSTEMS

In this section we provide a brief overview of existing TTA tools. Table 2 shows the names of the reviewed systems (if not named, we provide the name of the authors), with the year of publication and a brief description of what the animation tool does. Our results show that the interest in TTA systems increased in the mid-2000s, but subsequently dimmed by the end of the first decade. Another surge was recorded around the mid and late 2010s.

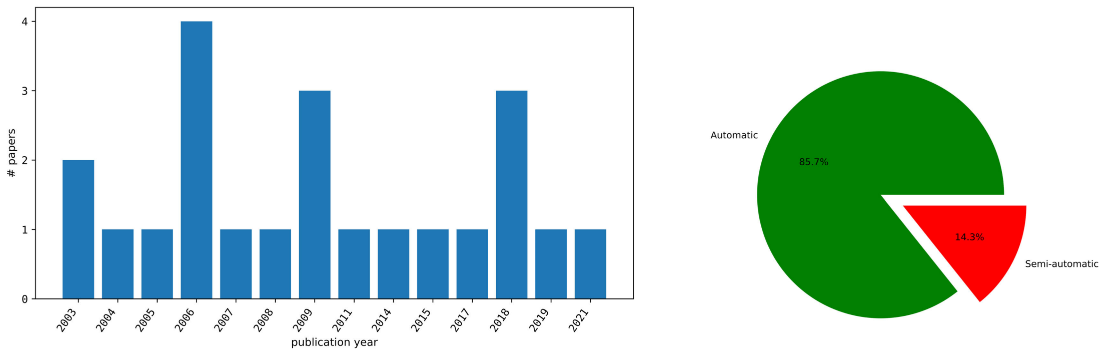
The 21 TTA systems reviewed in this study are of two types: automatic and semiautomatic. While the former allows the user to input natural language input and receive a 2D or 3D animation as an output, the latter allows the user to make modifications in at least one stage of the conversion. SWAN [7], for instance, allows the user to correct the output of the semantic parsing task and to intervene in the camera planning, light and color planning stages, while CONFUCIUS allows only interaction before the animation is generated [4]. Our results show that 18 of the surveyed systems do not allow the user to perform any changes to the output, restricting the interaction with the system to the NL input, whilst three systems allow the user to make modifications throughout the various stages of the conversion process.

The animation tools surveyed have been developed for a variety of topics, including instructional videos for exercising or cooking or to generate simulations, with storytelling, applied in children education and theatrical/cinematographic preproduction systems, dominating. Figure 6(a) shows that other domains like healthcare, sports and television have also received a fair share of interest from researchers.

Figure 6(b) shows that more than three quarters of the surveyed tools input English sentences, while the rest of the systems use either Japanese, Chinese or Swedish.

Figure 7 (a) shows that most animation tools produce 3D animations from NL input. Interest in 2D animation has dimmed since 2006, with the exception of text-to-dialogue [26]. The decreasing interest in 2D can be explained by the scarcity of ready-to-use resources needed to generate the animation. 3D resources, models and animations, are available in abundance online.

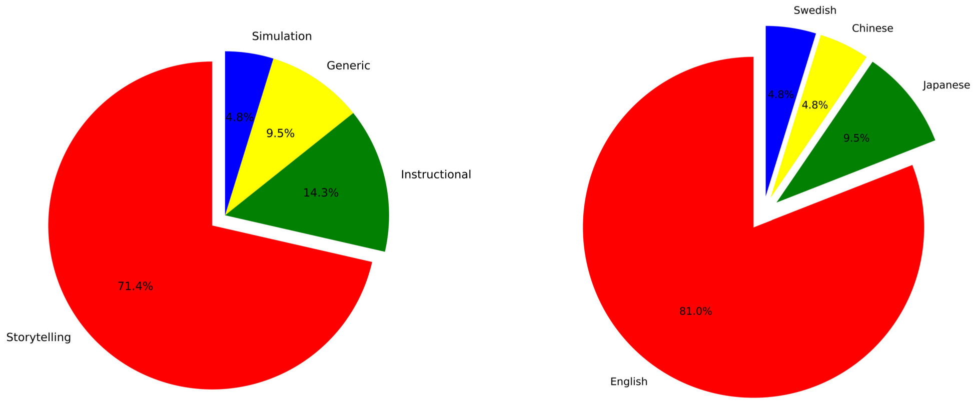
Figure 7 (b)also shows that two thirds of the animators restrict the input to single sentences, while the remaining seven systems allow the user to input multiple sentences. Such a design choice entails that the developers either enforce a template on the user to use a linear narrative or provide



(a) Publication Year of TTA Systems

(b) Types of TTA Systems

FIGURE 5. Types and publication year of TTA systems.



(a) Domains of TTA Systems

(b) Input Languages of TTA Systems

FIGURE 6. Domains and input of TTA systems.

a mechanism for temporal reasoning able to deal with non-linear narratives.

The issue of linear or non-linear narratives is nonexistent in systems that deal with imperative sentences, whereas declarative sentences may generate this issue. It is expected that the commands or instructions expressed using imperative sentences are linearly ordered. Figure 7 (c) shows that while three systems deal with imperative sentences, the rest of the systems uses declarative sentences.

The following section briefly describes each system in Table 2 from an NLP/NLU perspective, using the same chronological order.

B. NLP/NLU PERSPECTIVE

To assign the visual semantic roles to the input tokens, most systems adopt a rule-based approach which starts by syntactically parsing the input sentences. Accordingly, in this section, we look at how the TTA systems perform the syntactic parsing, how they assign the visual roles to the leaves of the parse trees and what knowledge base or bases they use to augment the extracted information before rendering the animation to the users.

Table 3 summarizes the findings of this section.

1) CARSIM

Carsim [27] initiates the NLU process by using regular expressions to identify verb patterns in the input. The process is followed by using a syntactic parser to extract the dependents of the verb. Carsim then uses Wordnet [28] to classify the extracted words according to a Wordnet internal hierarchy. The output of these sub-processes is then fed to an XML template which organizes it in three classes of elements: static objects, dynamic objects and collisions.

2) SWAN

SWAN [7], which was built to illustrate the feasibility of the FLICA methodology, uses a small subset of the Chinese language based on children stories and referred to as Moon Light. The subset has been made large enough that a large class of children stories can be created with it. Stories fed to the SWAN system are first checked for commonsense using the CSU grammar, which is short for CommonSense oriented Unification grammar. A context sensitive parser is then invoked to parse the Moon Light input and then match it

TABLE 2. Overview of TTA systems.

ID	System Name or Authors	Year	Brief Description
1	Carsim	2003	Converts written accident descriptions into animations to help insurance companies assess the reports.
2	SWAN	2003	SWAN converts natural language stories into cartoon, it allows users to intervene during the conversion process to accommodate the changes as they see fit.
3	Web2TV/Web2Talkshow	2004	These 2 complementary systems augment TV programs by fetching content from the web and then display it on TV, either as audio speech or using animated characters having a humorous dialogue.
4	e-hon	2006	Described as both a storytelling system and a multimedia communication tool, e-hon is a system that takes NL text and converts it into easily understandable storybook style with animation and dialogue in aims to help children understand difficult content.
5	ScriptViz1.0	2006	Converts movies scripts into 3D animations with a focus on the affective perspective.
6	CONFUCIUS	2006	Converts single sentences into 3D animations with audio and non-audio speech.
7	NLP StoryMaker	2006	Converts single sentences into 2D Animations.
8	Text2Dialogue(T2D)	2007	Converts natural language text into a dialogue and then has it animated using 3D characters.
9	CAMEO	2008	Takes screenplays as an input and converts it into 3D animations.
10	SceneMaker	2009	Converts movies scripts into 3D animations with a focus on the affective perspective.
11	Oshita	2009	This system converts NL movie scripts or stories into 3Danimations.
12	Web2Animation	2009	Converts cooking recipes on webpages to 3D animations.
13	Vist3D	2011	A system which allows for the creation of historical 3D spatiotemporal visualisations from natural language narratives.
14	Text-to-vision	2014	Converts natural language text on webpages into a TV-like program.
15	Chimmalgi	2015	Converts instructions of body exercises to 3D animations to help users perform exercises correctly.
16	Siddle et al.	2017	A system that finds application in the healthcare domain. It converts textual behavior recommendations into 3D animations to help patients better understand the actions they should do post-surgery.
17	AUI StoryMaker	2018	Converts single sentences into 3D animations with non-audio speech.
18	Cardinal	2018	Disney's text-to-VR system converts NL text from movie scripts to visualizations, in an attempt to reduce production time and cost.
19	TAPE	2018	Converts exercise descriptions into 3D animation to help people better understand physical exercise.
20	Zhang et al.	2019	Developed by Disney Research, it converts screenplays into animations, and extends Cardinal by adding a module of sentence simplification that gives the system the ability to handle complex sentences.
21	Mashad and Hamed	2021	Converts the text input into a 3D cartoon-like animation, it allows the users to choose the scene and the characters involved prior to generating the 3D output.

to a series of case frame CFs known as Golden Forest. SWAN uses one case frame per sentence and assumes that these are linear in the input, subsequently it disallows flashbacks without using special keywords to indicate them. Parsing and commonsense checking phases feed the CF with information on characters (as roles), objects, environments, and the actions involved, as described in the Moon Light sentence. SWAN invokes two different knowledge bases, Pangu for commonsense reasoning, and SWANLAKE which is a professional knowledge base geared towards director, lighting and color planning in animation generation systems.

3) Web2TV/Web2Talkshow

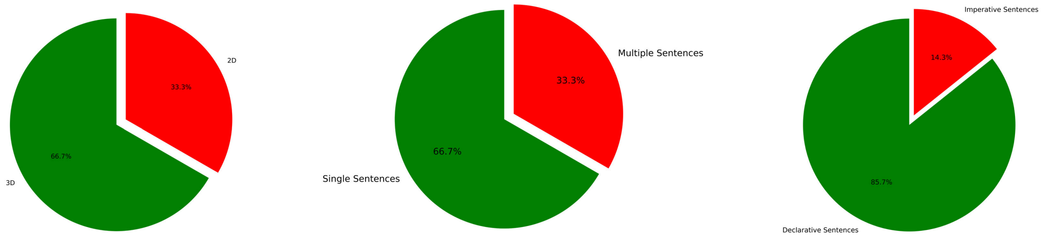
The idea behind Web2TV/Web2Talkshow [29] is to augment TV programs with content fetched from the web. The system was designed to convert content from web pages into a TV show style format, simply by showing the media on screen and using a Text-to-Speech (TTS) to convert the text into an audio modality. Another feature in the system allows the textual content in the fetched web pages to be converted into dialogue, which is then rendered on screen by two virtual agents. To convert a web page content into dialogue, a rule-based approach is adopted to extract the subject and content of the web page. The subject or subjects are extracted by looking at the keywords with the highest frequencies, and the corresponding contents are those terms with a high co-occurrence relationship with a specific subject term in the page. Human intervention is then allowed to create an XML template for the dialogue. To render the animated dialogue to the user, the system relies on TV program Making Language (TVML) [30].

4) E-HON

E-hon [31] uses two predefined tables for animation and background, against which it checks the information, which is extracted using a dependency parser, Cabocha [32], and tagged using semantics tags of time, space, weather and object. E-hon identifies the semantic categories with the use of a morphological analyzer and a Japanese lexicon. Once the tagged information has been checked against the aforementioned tables, an animation is invoked from a list of registered animations. An ontology is used to further explain texts presented as dialogues, for instance the system can convert "Antatanarivo in Madagascar" to "The city of Antatanarivo in the nation of Madagascar". As E-hon targets simplifying concepts to children, the use of the ontology helps clarify concepts to the users.

5) ScriptViz 1.0

Using the Applie Pie Parser [33], ScriptViz 1.0 [34] starts by syntactically analyzing the input. This phase outputs a parse tree which is analyzed to extract the animation elements. ScriptViz 1.0 uses a high-level planning module to convert the semantic information into a plan of action, which is represented as Parameterized Action Representation PAR.



(a) Output (b) Size of Input (c) Sentence Type

FIGURE 7. Input size and type and output of TTA systems.

6) CONFUCIUS

CONFUCIUS [35] and its successor SceneMaker rely on a syntactic parser (Connexor's Functional Dependency Grammar parser [36]) to parse the input, then extract the visual elements from the resulting parse tree. They can extract up to three elements in visual valency. For instance, a sentence like ''John gave Sarah a book'', defines an action verb ''Give'', with three visual valency elements:''John'', ''Sarah'' and ''Book''. A simpler input like ''John laughs'' identifies one visual valency element ''John'' for the action verb ''laugh''. The systems fill a Lexical Conceptual Structure (LCS) template with the extracted information, before passing the output in Virtual Reality Markup Language (VRML) to the graphics engine [37].

7) NLP STORY MAKER

Microsoft's NLP Story Maker [38] is comprised of a NLU module and a graphics engine. To extend its vocabulary, NLP Story Maker uses WordNet [28] to take into account synonyms of the actions supported by the system. The authors do not however report on the details of the semantic parsing or the rendering engine.

8) Text2Dialogue (T2D)

Text2Dialogue (T2D) [26], uses a different approach than a typical TTA system. Similarly to E-hon, it converts any type of text to a dialogue, which can then be performed by two virtual agents. T2D uses a Discourse Analyzing System (DAS), which builds Rhetorical Structure System (RST) structures. A mapper is then invoked, able to map these structures to DialogueNet Structures, before a presenter module translates them into Multimodal Presentation Markup Language (MPML3D) formal script [39], which can then be performed by the 3D agents.

9) CAMEO

CAMEO [40] takes an easier approach to generating an animation, in the sense that it asks the user for three inputs. The UserScript schema stores the scenarios. The ScreenShot schema contains information about 3D models to be shown in the resulting animation, including settings, characters, props, alongside the lights and cameras. The MediaStyle schema defines the genre, rhythm, atmosphere, and actors' characteristics. The system relies on a set of XML schemas to represent the different types of inputs necessary for 3D animation generation, and combines all three different XML files in a larger XML file referred to as the SceneScript. CAMEO's contribution is a knowledge base on direction techniques which was accumulated through conversations with real world experts.

10) SceneMaker

SceneMaker [5] is the successor of the CONFUCIUS TTA system. It extends it by adding emotional analysis and expressions through the use of Wordnet-Affect. Like CONFUCIUS, the system uses the Machinese Connexor POS Tagger, and type dependency analysis to identify the visual roles in the input sentence. Architecturally, SceneMaker exploits a client/server architecture, wherein the server is tasked with the NLU tasks, and the client does the rendering for the users.

11) OSHITA'S SYSTEM

Oshita's system [41] initiates the NLU process by invoking a syntactic parser, which converts plain text to a tree structure with phrases tags and dependencies. A semantic analysis phase is then started to extract the information about motions described in the input text from the tree structure. A Query Frame (QF) indicates which information the ''motion search'' should fetch from the ''motion database''. Temporal constraints are also extracted from the input text, and fed to the motion scheduling to determine the execution order of each motion captured as required by the QF.

12) Web2Animation

Web2Animation [42] narrows the vocabulary it works with to that of recipes. Web2Animation uses the Phoenix parser, a rule-based parser that relies on manually constructed semantic grammars [43]. The input is then converted to a sequence of semantic frames, in this case capturing the action, instrument and ingredient for each step in the recipe. A domain-specific ontology is then used to map the actions to the suitable graphical representation, and a user-specified dialogue is added to explain the recipe.

TABLE 3. NLU and knowledge bases used in TTA systems.

ID	System Name or Authors	Syntactic Analysis	Semantic Analysis	Knowledge Base
1	Carsim	Link Grammar dependency parser	Type Dependency Analysis	Wordnet
2	SWAN	Weak Precedence Story Parsing Grammar WPSPG	Golden Forest (Semantic Frames)	Pangu, Swanlake, CSU Grammar
3	Web2TV/Web2Talkshow	-	-	-
4	e-hon	Cabocha (Japanese dependency structure analyzer)	Role Labelling	Ontology
5	ScriptViz 1.0	Apple Pie Parser	Type Dependency Analysis	-
6	CONFUCIUS	Machinese Connexor	Type Dependency Analysis	Wordnet, LCSDatabase
7	NLP Story Maker	-	-	-
8	Text2Dialogue(T2D)	Machinese Syntax parser	Type Dependency Analysis	-
9	CAMEO	-	-	Directional knowledge database
10	Scene Maker	Machinese Connexor	Type Dependency Analysis	Wordnet, Wordnet-Affect, LCS Database and ConceptNet
11	Oshita	Stanford CoreNLP	Type Dependency Analysis	-
12	Web2Animation	Phoenix Parser	Type Dependency Analysis	Action Ontology
13	Vist3D	Regular Expressions	Role Labelling	-
14	Text-to-vision	-	-	-
15	Chimmalgi	Stanford Parser	Type Dependency Analysis	-
16	Siddle et al.	Bag of Words	-	PDDL Common Sense Knowledge and Reasoning
17	AUI Story Maker	OpenNLP	Type Dependency Analysis	-
18	Cardinal	Stanford Core NLP	Type Dependency Analysis	-
19	TAPE	Stanford Parser	Type Dependency Analysis	Bayesian Network
20	Zhang et al.	-	Role Labelling	-
21	Mashad and Hamed	Stanford CoreNLP	Type Dependency Analysis	Word2vec

13) Vist3D

Vist3D [44] exploits three elements to achieve its TTA capability. It first uses a narrative parser developed in PHP that relies on regular expressions to extract temporal information from the input and populate the temporal database. When no temporal information is left to be extracted, the narrative parser searches the input for basic sentence structures consisting of subject, verb and object. In the second stage, an analyser is used to create the scenario files by exploiting the Temporal Database (TD), the stored model and the terrain files to create the visualizations. The final stage allows rendering the animations in Panda3D [45] or VRML [37].

14) TEXT-TO-VISION

Text-to-vision [46] relies on the Flexible Interpretation Loader (FIL) as an intermediary between the natural language input and the rendered animation, details on how FIL performs the conversion are however omitted.

15) CHIMMALGI'S SYSTEM

Chimmalgi's system [47] uses the Stanford Parser [48] to generate a parse tree for input sentences. A dependency analysis takes place to determine the actions and the involved body parts as described in the sentence, and referred to in the system as ActionInfos. A matchscore is calculated based on bag-of-word approach, where common words between actioninfos are counted, added together and averaged. In a database of animations and models, an animation search is performed to retrieve the matching animations and models before being rendered to the user in a Unity-based viewer.

16) SIDDLE ET AL. SYSTEM

Siddle's system [49] starts the TTA process by using a deontic analysis able to capture forbidden and recommended deontics in sentences. An action recognition stage is then performed where a number of Finite States Transition Networks (FSTNs) are run against the sentences to extract actions, slot values (state of patient) and special instances. Common sense knowledge is then invoked to add information on the basic physics of the actions extracted in the previous stage. In the pre-final stage of the animation process, a mapping is created between the actions identified in stage 2, and the default action of the template, which results in the default template being updated to match the output of stages 2 and 3. Finally, the modified template is passed to the Unreal animation engine, which contains parametrized scripts updated according to the provided templates.

17) AUI STORY MAKER

AUI Story Maker [50] uses OpenNLP for syntactic parsing, it then analyzes the parse tree to identify the visual roles in the input sentence using Rusu's triplet extraction algorithm [51]. Similarly to CONFUCIUS and SceneMaker, AUI Story Maker uses an LCS template to further extract elements missing from the triplet extraction stage before submitting the resulting analysis to a Unity-based engine for rendering the output to the users.

18) CARDINAL

Developed by Disney Research, Cardinal [52] uses Stanford CoreNLP to perform a semantic analysis of the action text. It performs co-reference resolution first, then the main parsing task starts by extracting relation triples, subjects, relations (verbs) and objects. Cardinal is also able to extract verb modifiers, such as adverbs. The captured information is used to create affordances, which define a possible action in the TTA system. Architecturally, a segregation between the graphics and NLU engines is provided using a client/server architecture, so the input is sent for processing to the server, which transmits back the extracted triple. If the subject is available in the graphics engine, the corresponding affordance is retrieved. Cardinal uses the ADAPT framework based on the Unity 3D game engine to render the animation to the script writers.

19) TAPE

TAPE [53] begins the NLU process by using Stanford Parser for the syntactic parsing stage. It then uses custom rules fed to the Stanford parser for semantic information extraction, from which custom frames are filled with relevant information on actions, involved body parts and destination. To reason about the feasibility of the actions captured in the semantic analysis stage, a Bayesian network is used to infer the hidden information not explicitly denoted in the input case frames. The result of this stage is forward to the Artificial Social Agent Platform ASAP in Behavior Markup Language BML format and the animation is generated.

20) ZHANG ET AL. SYSTEM

Zhang's system [54] extends Cardinal by adding a sentence simplification module, making the system able to handle complex sentences. The process of sentence simplification is rule-based and relies on a Spacy-based dependency parser [55]. The sentence simplifier has two components, *"Identify"* which checks whether a given sentence matches a predefined grammatical structure, and a *"Transform"* component which proceeds to perform the actual simplification. The system then invokes an information extraction module which fills the visual elements into a predefined key-value pair structure referred to as Action Representation Fields (ARF), inspired by Badler's Parametrized Action Representation (PAR) [56]. The information extraction module relies on a pre-trained semantic role labeler [57] that inputs the sentence and output the values for the ARF keys. Cardinal's animation pipeline is then invoked to render the output to the user.

21) MASHAD AND HAMED SYSTEM

The process of animation generation begins with Named Entity Recognition (NER) stage allowing the system to identify the characters involved [58]. A co-reference resolution stage follows in which the system decides the actors and props in each input sentence. The TTA then the proceeds to extract the Subject-Action-Object (SAO) triplet from each given sentence by analyzing the dependency trees returned by the Stanford Parser [48]. The system uses a word2vec-based similarity check to map the unsupported actions to the closest system-supported actions [59]. The SAO elements are then communicated to a Unity-based graphics engine which proceeds to render the animation.

C. GRAPHICS ANALYSIS

In this section, we present the various graphics engines that existing TTA systems use to render the animations. Our results show that earlier systems relied on low-level graphics libraries such as OpenGL, its Java wrapper GL4Java or their Microsoft equivalent Direct X SDK. Carsim [27] relied on a higher-level graphics API, namely, Java3D. Up to 2015, 2/3D markup languages such as VRML, TVML or MPML3D were also used to create the animations. These languages have specialized software able to parse and render the described characters and motions.

As game engines gained in popularity, TTA tools have shifted to rely on such technology. Game engines provide a separation between the characters, animations, backgrounds and props (referred to as assets) and the scripts that allow these elements to interact in the animation. It's also relatively easy to script an animation, provided access to these assets.

The majority of the systems we surveyed allow users to interact using NL only, and disallow any further interaction through the process of NLU and reasoning. The vast majority of the systems rely on a markup language that sits between the NLU and the graphics engine, which, in some cases, allows the users room to edit the output before it is rendered as animation to the user.

Table 4 summarizes the findings of this analysis.

V. DISCUSSION AND EVALUATION

Despite the numerous research works available in the field of TTA systems, none of them are available online, and none have been widely used in any of the domains they have been designed for. The same cannot be said for text-to-scene systems. For example, WordsEye provides a web application that the general public can use to create 3D scenes [1]. The complexity of the challenges underlying TTA systems is however higher than that of text-to-picture or text-to-scene tools, as it encompasses temporal reasoning challenges and spatio-temporal reasoning (objects position through time), in addition to the study of spatial relationships.

A. DOMAINS AND LANGUAGES

Amongst the surveyed animation tools, 90% have been targeted towards a specific domain, either storytelling or instructional animations. Given that TTA systems rely on a database of graphics used to instantiate the required objects into the 2D or 3D scene, the restriction of the context is crucial to the success of the animation process. Restricting the context however does not necessarily entail the restriction of the vocabulary, as (some of) the NLU modules may rely on lexical databases, such as Wordnet, which are able to provide synonyms for words not included in the vocabulary

TABLE 4. Graphics analysis of TTA systems.

ID	System Name or Authors	Markup Language	Graphics Engines or Library
1	Carsim	XML	Java3D
2	SWAN	XML	Unknown
3	Web2TV/Web2Talkshow	TVML	TVML Player
4	e-hon	None	Direct X SDK
5	ScriptViz 1.0	-	GL4Java (OpenGL for Java)
6	CONFUCIUS	VRML	VRML Viewer
7	NLP Story Maker	XML	Unknown
8	Text2Dialogue(T2D)	Multimodal Presentation Markup Language 3D (MPML3D)	OpenGL
9	CAMEO	XML	Unknown
10	Scene Maker	VRML	VRML Viewer
11	Oshita	XML	Smart Motion Synthesis
12	Web2Animation	XML	Unknown
13	Vist3D	VRML	Web Browser or Panda 3D
14	Text-to-vision	TVML	TVML Player
15	Chimmalgi	XML	Unity 3D
16	Siddle et al.	-	Unreal Development Kit UDK
17	AUI StoryMaker	XML	Unity 3D
18	Cardinal	-	ADAPT (Unity 3D)
19	TAPE	Behavior Markup Language BML	Artificial Social Agent Platform (ASAP)
20	Zhang et al.	Action Representation Fields (ARF)	ADAPT (Unity 3D)
21	Mashad and Hamed	-	Unity 3D

of the TTA tool. We will argue here that the grammatical structures are not domain dependent, despite the differences in the grammar used for example by a child and that of an adult. With the exception of Carsim, which handles input in three languages, Swedish, English and French, the remaining TTA systems are monolingual. A work-around solution for multilingualism could be to use a translation module before the NLU intercepts the input. This may generate some problems if the translation is not accurate, further complexifying the work of the TTA tool.

B. NATURAL LANGUAGE UNDERSTANDING

Around 60% of the surveyed systems rely on a syntactic parser that outputs a parse tree where syntactic roles are later replaced by semantic roles. Another 20% of the TTA systems skips the syntactic parsing and adopts more basic information extraction approaches such as bag of words or regular expressions, before assigning relevant candidate tokens visual semantic roles in their targeted visualizations. While it is hard to evaluate how well these approaches work in extracting the visual semantic roles from the natural language input, we know that the NLU stage itself is not enough to produce an adequate animation for an input text. The process has to be augmented with visual reasoning to address the plethora of issues we discussed in Section III. In that, 30% of the surveyed systems overlook visual reasoning as a whole, and choose to lay the task of completing the animations on the shoulder of the user. The remaining systems attempt to address various problems at the reasoning level, either by relying on generic ontologies or knowledge bases (Wordnet, ConceptNet, PDDL, etc.), by developing animation-specific knowledge bases such as Pangu, SwanLake, or even through the use of Bayesian networks.

C. INPUT SIZE AND INTERACTIVITY OF TTA SYSTEMS

It is difficult to map natural language texts to the adequate animations, this is made especially more complex with the ‘‘underspecification’’ problem discussed in Section III, and the ambiguity usually tied to natural language. This stresses the need for more user-centered designs of TTA systems, which has not been adopted in 85% of the surveyed systems, as users are not allowed in 18 systems to alter the output of the sub-processing in the pipelines of TTA systems. Such user involvement is possible in at least 15 of the surveyed systems, as these rely on a markup language which transmits the animation data from the NLU module to the graphics engines or libraries. These markup languages are human-readable, and at the exception of the two systems designed for children, [38] and [50], the users can easily alter any mistakes made by the NLP modules before this is forwarded to the graphics engine for rendering. Another limitation these systems raise relate to the input size, as two thirds of the surveyed systems allow only single-sentence input, rendering the process of animation generation rather slow, and time-consuming, as for example in digital storytelling, the users will have to decompose the story into sentences first, and re-order them to ensure the linearity of the narrative.

D. AN EVALUATION OF THE TTA SYSTEMS OVER THE CHALLENGES

We have studied the 21 reviewed systems and looked at how they address the three classes of challenges we identified in Section III: NLU, temporal and visual reasoning. We looked at the various mechanisms and algorithms these systems have provided to solve the identified challenges. We assigned a binary rating to each of these algorithms. This allowed us to gain an overview of how broadly the reviewed systems

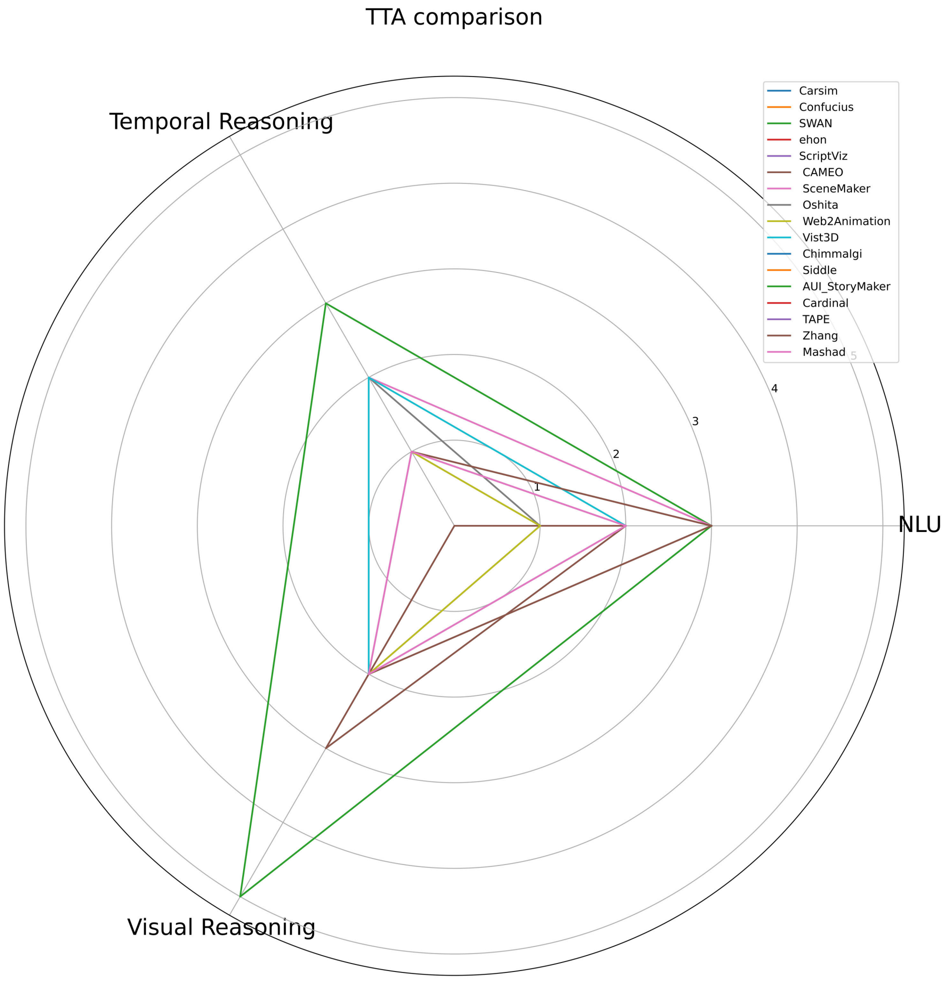


FIGURE 8. **TTA systems performance on the challenges.**

deal with TTA challenges. Figure 8 shows the results of the evaluation. Only one system, SWAN, attempted to solve the five challenges in visual reasoning, though it only dealt with two challenges in temporal reasoning and three on the NLU side. The remaining systems have concentrated on solving the most basic issues on every dimension. This assessment testifies to the complexity of the problem itself, as each of the dimensions is composed of five problems (see Figure 4) each of which builds on the other and requires much research to be solved, in this context, on its own.

Our assessment is however biased towards TTA systems designed for storytelling, as we do not expect a system that generates animations from cooking recipes or physical exercise descriptions to tackle the issues related to temporal reasoning. Such systems do not handle inputs with temporal expressions, or inputs where the verb tenses might suggest a non-linear narrative. We expect also that quantification problems and camera planning are not relevant issues for systems handling instructional texts or systems that output the NL input as a dialogue between virtual characters.

VI. CONCLUSION

By setting some rigorous inclusion criteria, we have been able to isolate systems that generate animations from natural language input, focusing on those that perform some natural language understanding and are not purely data-driven.

Our findings can be summarized as follows: Despite a space for user interaction throughout the conversion process, most systems chose to conclude the process without any user intervention. Such a limitation, can result in the wrong animation being generated, worsened by the inherent ambiguity of natural language.

The visual semantic parsing task has proven to be rule-based in most of the reviewed systems, despite its reliance on statistical tools for either syntactic parsing or part-of-speech tagging.

The core of the problem of animation generation goes beyond that of semantic parsing, and lies, as discussed, in visual reasoning, a task that is made more complex by the problem of underspecification. While a number of systems deal with the issue through various knowledge bases and ontologies, or in some cases Bayesian networks, other systems choose to hardcode the visual reasoning in the body of the animator.

Among our recommendations for TTA systems is closing the domain of the system and its vocabulary, in order to address the issue of the absence of props, actions or actors in the animator's database. TTA systems should also offer a space for the users wherein they could fix or supplement the output of the NLU process. The use of a more human-readable language sitting between the NLU engine and the graphics engine will allow less tech-savvy people to intervene in the animation process.

REFERENCES

[1] A. Chang, M. Savva, and C. Manning, "Semantic parsing for text to 3D scene generation," in *Proc. ACL Workshop Semantic Parsing*, 2014, pp. 17–21.

[2] M. B. Islam, A. Ahmed, M. K. Islam, and A. K. Shamsuddin, "Child education through animation: An experimental study," 2014, *arXiv:1411.1897*.

[3] M. Jancheski, "The importance of animations and simulations in the process of (e-)learning," in *Proc. 8th Conf. Inform. Inf. Technol. Int. Participation (CIIT)*, 2011, pp. 177–191.

[4] M. Ma, "Automatic conversion of natural language to 3D animation," Ph.D. dissertation, Dept. Eng., Univ. Ulster, Coleraine, U.K., 2006.

[5] E. Hanser, P. M. Kevitt, T. Lunney, and J. Condell, "SceneMaker: Automatic visualisation of screenplays," in *Proc. Annu. Conf. Artif. Intell.* Cham, Switzerland: Springer, 2009, pp. 265–272.

[6] K. Hassani and W.-S. Lee, "Visualizing natural language descriptions: A survey," *ACM Comput. Surv.*, vol. 49, no. 1, pp. 1–34, Mar. 2017.

[7] R. Lu and S. Zhang, *Automatic Generation of Computer Animation: Using AI for Movie Animation*, vol. 2160. Cham, Switzerland: Springer, 2003.

[8] J. Hou, X. Wang, F. Xu, V. D. Nguyen, and L. Wu, "Humanoid personalized avatar through multiple natural language processing," *Int. J. Comput. Inf. Eng.*, vol. 3, no. 11, pp. 2731–2736, 2009.

[9] R. Winkler, S. Hobert, A. Salovaara, M. Söllner, and J. M. Leimeister, "Sara, the lecturer: Improving learning in online education with a scaffolding-based conversational agent," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2020, pp. 1–14.

[10] B. Kitchenham, "Procedures for performing systematic reviews," Dept. Comput. Sci., Keele Univ., Keele, U.K., Tech. Rep. 2004, pp. 1–26.

[11] J. Zakraoui, M. Saleh, and J. A. Ja'am, "Text-to-picture tools, systems, and approaches: A survey," *Multimedia Tools Appl.*, vol. 78, no. 16, pp. 22833–22859, Aug. 2019.

[12] A. S. Lin, L. Wu, R. Corona, K. W. H. Tai, Q. Huang, and R. J. Mooney, "Generating animated videos of human activities from natural language descriptions," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–7.

[13] T. Yamada, H. Matsunaga, and T. Ogata, "Paired recurrent autoencoders for bidirectional translation between robot actions and linguistic descriptions," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3441–3448, Oct. 2018.

[14] M. Plappert, C. Mandery, and T. Asfour, "Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks," *Robot. Auto. Syst.*, vol. 109, pp. 13–26, Nov. 2018.

[15] Y. Pan, Z. Qiu, T. Yao, H. Li, and T. Mei, "To create what you tell: Generating videos from captions," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 1789–1798.

[16] T. Gupta, D. Schwenk, A. Farhadi, D. Hoiem, and A. Kembhavi, "Imagine this! Scripts to compositions to videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 598–613.

[17] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, vol. 3. 2014.

[18] J. Ruppenhofer and I. Rehbein, "Yes we can? Annotating the senses of English modal verbs," in *Proc. 8th Int. Conf. Lang. Resour. Eval. (LREC)*, European Language Resources Association, 2012, pp. 1538–1545.

[19] Z. Vendler, "Verbs and times," *Phil. Rev.*, vol. 66, no. 2, pp. 143–160, 1957.

[20] C. Hashimoto, K. Torisawa, K. Kuroda, S. De Saeger, M. Murata, and J. Kazama, "Large-scale verb entailment acquisition from the web," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2009, pp. 1172–1181.

[21] L. Ruqian, Z. Songmao, H. Shi, and Y. Lixin, "The quantification problem in animation generation," in *Proc. 3rd Austral. New Zealand Conf. Intell. Inf. Systems. (ANZIIS)*, 1995, pp. 93–98.

[22] L.-W. He, M. F. Cohen, and D. H. Salesin, "The virtual cinematographer: A paradigm for automatic real-time camera control and directing," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, 1996, pp. 217–224.

[23] D. Amerson, S. Kime, and R. M. Young, "Real-time cinematic camera control for interactive narratives," in *Proc. ACM SIGCHI Int. Conf. Adv. Comput. Entertainment Technol.*, Jun. 2005, p. 369.

[24] I.-C. Yeh, C.-H. Lin, H.-J. Chien, and T.-Y. Lee, "Efficient camera path planning algorithm for human motion overview," *Comput. Animation Virtual Worlds*, vol. 22, nos. 2–3, pp. 239–250, Apr. 2011.

[25] A. Amamra, Y. Amara, R. Benaissa, and B. Merabti, "Optimal camera path planning for 3D visualisation," in *Proc. Comput. Conf. (SAI)*, Jul. 2016, pp. 388–393.

[26] P. Piwek, H. Hernault, H. Prendinger, and M. Ishizuka, "T2D: Generating dialogues between virtual agents automatically from text," in *Proc. Int. Workshop Intell. Virtual Agents*, Paris, France, 2007, pp. 161–174.

[27] O. Åkerberg, H. Svensson, B. Schulz, and P. Nugues, "CarSim: An automatic 3D text-to-scene conversion system applied to road accident reports," in *Proc. Demonstrations*, 2003, pp. 191–194.

[28] C. Fellbaum, "WordNet: An electronic lexical resource," *The Oxford Handbook of Cognitive Science*. Oxford Univ. Press, 2017, ch. 16, p. 301.

[29] A. Nadamoto and K. Tanaka, "Complementing your TV-viewing by web content automatically-transformed into TV-program-type content," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, Nov. 2005, pp. 41–50.

[30] M. Hayashi, "Machine TV program generation from text-based script," in *Proc. 2nd Symp. Intell. Inf. Media*, 1996, pp. 137–144.

[31] K. Sumi and M. Nagata, "Animated storytelling system via text," in *Proc. ACM SIGCHI Int. Conf. Adv. Comput. Entertainment Technol.*, Jun. 2006, p. 55.

[32] T. Kudo and Y. Matsumoto, "Japanese dependency analysis using cascaded chunking," in *Proc. 6th Conf. Natural Lang. Learn. (COLING)*, 2002, pp. 1–7.

[33] S. Sekine and R. Grishman. (1996). *The Apple Pie Parser*. [Online]. Available: http://www.cs.nyu.edu/cs/projects/proteus/app/

[34] Z.-Q. Liu and K.-M. Leung, "Script visualization (ScriptViz): A smart system that makes writing fun," *Soft Comput.*, vol. 10, no. 1, pp. 34–40, Jan. 2006.

[35] M. Ma and P. M. Kevitt, "Visual semantics and ontology of eventive verbs," in *Proc. Int. Conf. Natural Lang. Process.*, 2004, pp. 187–196.

[36] P. Tapanainen and T. Järvinen, "A non-projective dependency parser," in *Proc. 5th Conf. Appl. Natural Lang. Process.*, 1997, pp. 64–71.

[37] R. Carrey and G. Bell, *The Annotated VRML 2.0 Reference Manual*. Reading, MA, USA: Addison-Wesley, 1999.

[38] T. Aikawa, L. Schwartz, and M. Pahud, "NLP story maker," in *Proc. 2nd Lang. Technol. Conf., Human Lang. Technol. Challenge Comput. Sci. Linguistics*, 2005, pp. 1–4.

[39] Z. Yang and M. Ishizuka, "MPML-FLASH: A multimodal presentation markup language with character agent control in flash medium," in *Proc. 24th Int. Conf. Distrib. Comput. Syst. Workshops*, 2004, pp. 537–543.

[40] H. Shim and B. G. Kang, "CAMEO—Camera, audio and motion with emotion orchestration for immersive cinematography," in *Proc. Int. Conf. Adv. Comput. Entertainment Technol.*, Dec. 2008, pp. 115–118.

[41] M. Oshita, "Generating animation from natural language texts and framework of motion database," in *Proc. Int. Conf. CyberWorlds*, 2009, pp. 146–153.

[42] H. Shim, B. Kang, and K. Kwag, "Web2Animation—Automatic generation of 3D animation from the web text," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, Sep. 2009, pp. 596–601.

[43] W. Ward, "Understanding spontaneous speech: The Phoenix system," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1991, pp. 365–367.

[44] A. Oddie, P. Hazlewood, B. Farrimond, and S. Presland, "Applying deductive techniques to the creation of realistic historical 3D spatiotemporal visualisations from natural language narratives," in *Proc. Electron. Visualisation Arts (EVA)*, 2011, pp. 97–105.

[45] M. Goslin and M. R. Mine, "The Panda3D graphics engine," *Computer*, vol. 37, no. 10, pp. 112–114, Oct. 2004.

[46] M. Hayashi, S. Inoue, M. Douke, N. Hamaguchi, H. Kaneko, S. Bachelder, and M. Nakajima, "T2 V: New technology of converting text to CG animation," *ITE Trans. Media Technol. Appl.*, vol. 2, no. 1, pp. 74–81, 2014.

[47] R. V. Chimmalgi, "Automated conversion of text instructions to human motion animation," Agricult. Mech. College, Louisiana State Univ., Baton Rouge, LA, USA, Tech. Rep. 2015.

[48] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proc. 41st Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2003, pp. 423–430.

[49] J. Siddle, A. Lindsay, J. F. Ferreira, J. Porteous, J. Read, F. Charles, M. Cavazza, and G. Georg, "Visualization of patient behavior from natural language recommendations," in *Proc. Knowl. Capture Conf.*, Dec. 2017, pp. 1–4.

[50] N. Bouali and V. Cavalli-Sforza, "AUI story maker: Animation generation from natural language," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2018, pp. 424–428.

[51] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic, "Triplet extraction from sentences," in *Proc. 10th Int. Multiconf. Inf. Soc. (IS)*, 2007, pp. 8–12.

[52] M. Marti, J. Vieli, W. Witoń, R. Sanghrajka, D. Inversini, D. Wotruba, I. Simo, S. Schriber, M. Kapadia, and M. Gross, "CARDINAL: Computer assisted authoring of movie scripts," in *Proc. 23rd Int. Conf. Intell. User Interfaces*, Mar. 2018, pp. 509–519.

[53] H. Sarma, R. Porzel, J. D. Smeddinck, R. Malaka, and A. B. Samaddar, "A text to animation system for physical exercises," *Comput. J.*, vol. 61, no. 11, pp. 1589–1604, 2018.

[54] Y. Zhang, E. Tsipidi, S. Schriber, M. Kapadia, M. Gross, and A. Modi, "Generating animations from screenplays," 2019, *arXiv:1904.05440*.

[55] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," vol. 7, no. 1, pp. 411–420, 2017.

[56] N. I. Badler, R. Bindiganavale, J. Allbeck, W. Schuler, L. Zhao, and M. Palmer, "Parameterized action representation for virtual human agents," in *Embodied Conversational Agents*. Cambridge, MA, USA: MIT Press, 2000, pp. 256–284.

[57] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. S. Zettlemoyer, "AllenNLP: A deep semantic natural language processing platform," 2017, *arXiv:1803.07640*.

[58] S. Y. El-Mashad and E.-H.-S. Hamed, "Automatic creation of a 3D cartoon from natural language story," *Ain Shams Eng. J.*, vol. 13, no. 3, May 2022, Art. no. 101641.

[59] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient representation of word representations in vector space," in *Proc. Int. Workshop Learn. Represent. (ICLR)*, 2013, pp. 1–12.

NACIR BOUALI received the bachelor's degree in software design from Université Hassan 1er de Settat, in 2011, and the master's degree in software engineering from Al Akhawayn University in Ifrane, in 2016. He is currently pursuing the Ph.D. degree in virtual reality generation from natural language with the University of Eastern Finland.

He is a Lecturer with the University of Twente, The Netherlands, where he teaches courses on artificial intelligence, machine learning, and databases. His research interests include AI in education, natural language processing, and the use of immersive virtual environments in language learning.

VIOLETTA CAVALLI-SFORZA received the degree in civil engineering, the degree in computer science, and the Ph.D. degree in intelligent systems studies. Her Ph.D. dissertation focused on computer-assisted instruction to visualize scientific argumentation.

Since 2008, she has been an Associate Professor of computer science with Al Akhawayn University in Ifrane (AUI). She has also coordinated and contributed to the Faculty Development Center, AUI. She worked in natural language processing, particularly in machine translation, with Carnegie Mellon University, but her research in the last few years has focused on language learning through reading and dialogue and on readability of texts, with a focus on Arabic. She has taught a variety of topics in computer science at the graduate and undergraduate level.