Converting Texts of Road Accidents into 3D Scenes

Richard Johansson and **David Williams** and **Pierre Nugues**[1]

Abstract. This paper describes a system to create animated 3D scenes of car accidents from reports written in Swedish. The system has been developed using news reports of varying size and complexity. The text-to-scene conversion process consists of two stages. An information extraction module creates a tabular description of the accident and a visual simulator generates and animates the scene.

1 Carsim

Carsim [4] is a text-to-scene converter applied to road accidents reports. It analyzes texts describing accidents and visualizes them in a 3D environment. The Carsim architecture is divided into two parts that communicate using a formal representation of the accident. This formalism adopts a template structure similar to that of information extraction systems.

Carsim's first part is a linguistic module that extracts information from the report and fills the template slots. The second part is a virtual scene generator that takes the filled template as an input, creates the visual entities, and animates them.

The conversion of natural language texts into graphics has been investigated in a few other projects. NALIG [1], WordsEye [3], and CogViSys [2] are examples of it.

2 The Corpus

As development and test sets, we have collected approximately 200 reports of road accidents from various Swedish newspapers. The task of analyzing the news reports is made more complex by their variability in style and length. The size of the texts ranges from a couple of sentences to more than a page. The amount of details is overwhelming in some reports, while in others most of the information is implicit. The complexity of the accidents described ranges from simple accidents with only one vehicle to multiple collisions with several participating vehicles.

The next text is an example of a press wire describing an accident, which is part of our development corpus.

> Tre personer omkom när en buss och personbil på måndagen krockade på väg 55 vid Fornebo i närheten av Flen. Det var ett barn och två vuxna som färdades i personbilen som omkom i olyckan. Ytterligare ett barn, en flicka, fanns i bilen, men kunde ta sig ut. - Hon fick hjälp av en person att ta sig ut ur bilen, berättar Mats Elfwén, räddningsledare vid räddningstjänsten i Flen, för TT. Han vet inte hur olyckan gick till. - Av någon anledning kom personbilen över på fel sida med sladd. Bussföraren försökte undvika den, men det blev en frontalkollision, säger Mats Elfwén. Vid krocken fattade personbilen eld. Flickan som räddades ur bilen fördes till sjukhus med bland annat brännskador. Ungefär 15 personer från räddningstjänsterna i Flen och Malmköping deltog i arbetet vid olyckan.
>
> TT, November 11, 2002.

Three persons were killed last Monday when a bus and a passenger car collided on Road 55 at Fornebo close to Flen. The victims in the accident were a child and two adults who were traveling in the car. Another child, a girl, was in the car but managed to escape. – Someone helped her to get out of the car, Mats Elfwén, leader of the Flen emergency staff, told TT. He is not aware of how the accident took place. – For some reason the car skidded over to the wrong side of the lane. The bus driver could not avoid a frontal collision, said Mats Elfwén. During the collision, the car caught fire. The girl who was saved from the car was brought to hospital with burns. About 15 people from the Flen and Malmköping emergency staffs participated in the accident rescue.

3 Information Expressed by the Formalism

The Carsim language processing module reduces the text content to a tabular structure that outlines what happened and enables a conversion to a symbolic scene. It uses information extraction techniques to map a text onto a predefined XML structure.

The template corresponding to the example above contains:

- A scene element with the location of the accident and the configuration of roads;
- A list of road objects with two cars and one truck;
- An event chains for the objects with their movements: the second car overtakes the first car and gets into a skid.
- Finally, a collision list with a collision object describing the collision between the car and the truck, where the side of the car hits the front of the truck.

4 The Information Extraction Module

The information extraction (IE) subsystem fills the template slots. Its processing flow consists in analyzing the text linguistically and framing the accident into a prototype situation using the word groups obtained from the linguistic modules and a sequence of semantic modules. The IE subsystem uses the literal content of certain phrases it finds in the text or infers the environment and the actions.

We use a pipeline of modules in the first stages of the processing chain. The tasks consists of tokenizing, part-of-speech tagging, splitting into sentences, detecting the noun groups, named entities, and domain-specific multiwords. We also detect non-recursive clauses.

[1] Lund University, LTH, Box 118, S-221 00 Lund, Sweden, email: richard@cs.lth.se, d98dw@efd.lth.se, Pierre.Nugues@cs.lth.se

4.1 Detecting the Participants

The system extracts the noun groups and uses them to identify the participating road objects. They consist of two categories: the vehicles and the obstacles organized in a small ontology. The system extracts the headword of each group and associates it to an entity in the ontology.

We track the entities along the text with a simple coreference resolution algorithm. It assumes that each definite expression corefers with the last sortally consistent entity (according to the ontology), which was mentioned. Indefinite expressions are assumed to be references to previously unmentioned entities.

The excerpt below shows the annotation of the participants and the road objects as well as their coreferences:

> The accident took place when [the car]$_1$ where the five people were traveling overtook [another passenger car]$_2$. When [it]$_1$ turned back in front of [the overtaken car]$_2$, [it]$_1$ got into a skid and came with the side towards the front of [the meeting truck]$_3$.

4.2 Marking Up the Events

Events in car accident reports correspond to vehicle motions and collisions. We detect them to be able to visualize and animate the scene actions. We collected verbs and nouns depicting vehicle activity and maneuvers that we used to anchor the event identification and as well as their semantic roles to determine the event arguments.

The sentence below shows the groups marked up and labeled with their semantic roles.

> [About five]$_{Time}$ [on Thursday afternoon]$_{Time}$, [a passenger car]$_{Actor}$ **drove** [into a terrace house]$_{Victim}$ [in an old people's home]$_{Location}$ [at Alvägen street]$_{Location}$ [in Enebyberg]$_{Location}$ [north of Stockholm]$_{Location}$.

Carsim labels automatically semantic roles. It uses an algorithm to similar to [5]. However, as there is no lexical resource such as FrameNet for Swedish and no widely available parser, we adapted it. We used a more local strategy as well as a different set of learning attributes.

The analysis starts from the verbs and nouns in our ontology for which we designed a specific set of frames and associated roles. It limits the scope of each event to the clause where it appears and identifies the verb and noun dependents: noun groups, prepositional groups, and adverbs that it classifies according to semantic roles.

We annotated manually a set of 819 examples on which we trained and tested our classifier. We used a random subset of 10% of the examples as the test set and the rest as the development set. We repeated the experiment on different subsets and we kept the minimal value as the accuracy of the classifier: 90 percent.

When the events have been detected in the text, they can be interpreted and stored in the template structure. Carsim first maps the events directly onto the template. This produce correct results for short texts but results in overgeneration in long texts. The cause is often due to the multiple mentions of a same event. We remove duplicates that have the same semantic frame and that are filled with the same actors.

4.3 Detecting the Roads

The configuration of roads and the rest of the static scene is inferred from the information in the detected events. When one of the involved vehicles makes a turn, this indicates that the configuration is probably a crossroads.

Additional information is provided using keyword spotting in the text. Examples of relevant keywords are *korsning* ('crossing'), *rondell* ('roundabout'), and *kurva* ('bend'), which are very likely indicators of the road configuration if seen in the text.

5 Scene Synthesis and Visualization

The visualizer reads its input from the template description. It synthesizes a symbolic 3D scene and animates the vehicles.

The scene generation algorithm positions the static objects and plans the vehicle motions. It uses rule-based modules to check the consistency of the template description and to estimate the 3D start and end coordinates of the vehicles. The visualizer uses a planner to generate the vehicle trajectories and a temporal module assigns time intervals to all the segments of the trajectories.

Figure 1 shows two screenshots generated from the text above. It should be noted that the graphic representation is intended to be iconic in order not to convey any meaning which is not present in the text.

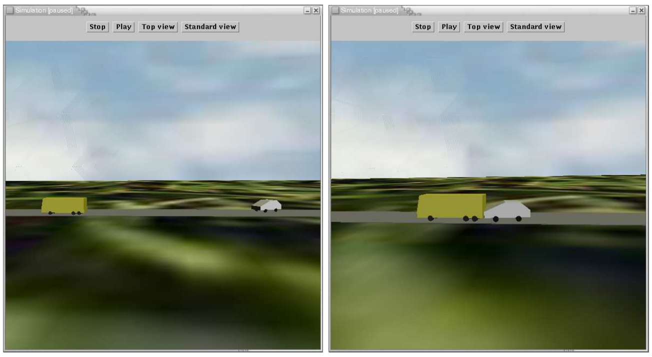

Figure 1. Screenshots from the animation of the text above.

ACKNOWLEDGEMENTS

This work is partly supported by grant number 2002-02380 from the Vinnova Språkteknologi program.

REFERENCES

[1] Giovanni Adorni, Mauro Di Manzo, and Fausto Giunchiglia, 'Natural language driven image generation', in *Proceedings of COLING 84*, pp. 495–500, Stanford, California, (1984).

[2] Michael Arens, Artur Ottlik, and Hans-Hellmut Nagel, 'Natural language texts for a cognitive vision system', in *ECAI2002, Proceedings of the 15th European Conference on Artificial Intelligence*, ed., Frank van Harmelen, Lyon, (July 21-26 2002).

[3] Bob Coyne and Richard Sproat, 'Wordseye: An automatic text-to-scene conversion system', in *Proceedings of the Siggraph Conference*, Los Angeles, (2001).

[4] Sylvain Dupuy, Arjan Egges, Vincent Legendre, and Pierre Nugues, 'Generating a 3D simulation of a car accident from a written description in natural language: The Carsim system', in *Proceedings of The Workshop on Temporal and Spatial Information Processing*, pp. 1–8, Toulouse, (July 7 2001). Association for Computational Linguistics.

[5] Daniel Gildea and Daniel Jurafsky, 'Automatic labeling of semantic roles', *Computational Linguistics*, **28**(3), 245–288, (2002).