

Generation of landmark-based navigation instructions from open-source data

Markus Dräger

Dept. of Computational Linguistics
Saarland University
mdraeger@coli.uni-saarland.de

Alexander Koller

Dept. of Linguistics
University of Potsdam
koller@ling.uni-potsdam.de

Abstract

We present a system for the real-time generation of car navigation instructions with landmarks. Our system relies exclusively on freely available map data from OpenStreetMap, organizes its output to fit into the available time until the next driving maneuver, and reacts in real time to driving errors. We show that female users spend significantly less time looking away from the road when using our system compared to a baseline system.

1 Introduction

Systems that generate route instructions are becoming an increasingly interesting application area for natural language generation (NLG) systems. Car navigation systems are ubiquitous already, and with the increased availability of powerful mobile devices, the wide-spread use of pedestrian navigation systems is on the horizon. One area in which NLG systems could improve existing navigation systems is in the use of *landmarks*, which would enable them to generate instructions such as “turn right after the church” instead of “after 300 meters”. It has been shown in human-human studies that landmark-based route instructions are easier to understand (Lovelace et al., 1999) than distance-based ones and reduce driver distraction in in-car settings (Burnett, 2000), which is crucial for improved traffic safety (Stutts et al., 2001). From an NLG perspective, navigation systems are an obvious application area for situated generation, for which there has recently been increasing interest (see e.g. (Lessmann et al., 2006; Koller et al., 2010; Striegnitz and Majda, 2009)).

Current commercial navigation systems use only trivial NLG technology, and in particular are

limited to distance-based route instructions. Even in academic research, there has been remarkably little work on NLG for landmark-based navigation systems. Some of these systems rely on map resources that have been hand-crafted for a particular city (Malaka et al., 2004), or on a combination of multiple complex resources (Raubal and Winter, 2002), which effectively limits their coverage. Others, such as Dale et al. (2003), focus on non-interactive one-shot instruction discourses. However, commercially successful car navigation systems continuously monitor whether the driver is following the instructions and provide modified instructions in real time when necessary. That is, two key problems in designing NLG systems for car navigation instructions are the availability of suitable map resources and the ability of the NLG system to generate instructions and react to driving errors in real time.

In this paper, we explore solutions to both of these points. We present the Virtual Co-Pilot, a system which generates route instructions for car navigation using landmarks that are extracted from the open-source OpenStreetMap resource.¹ The system computes a route plan and splits it into episodes that end in driving maneuvers. It then selects landmarks that describe the locations of these driving maneuvers, and aggregates instructions such that they can be presented (via a TTS system) in the time available within the episode. The system monitors the user’s position and computes new, corrective instructions when the user leaves the intended path. We evaluate our system using a driving simulator, and compare it to a baseline that is designed to replicate a typical commercial navigation system. The Virtual Co-Pilot performs comparably to the baseline

¹<http://www.openstreetmap.org/>

on the number of driving errors and on user satisfaction, and outperforms it significantly on the time female users spend looking away from the road. To our knowledge, this is the first time that the generation of landmarks has been shown to significantly improve the instructions of a wide-coverage navigation system.

Plan of the paper. We start by reviewing earlier literature on landmarks, route instructions, and the use of NLG for route instructions in Section 2. We then present the way in which we extract information on potential landmarks from OpenStreetMap in Section 3. Section 4 shows how we generate route instructions, and Section 5 presents the evaluation. Section 6 concludes.

2 Related Work

What makes an object in the environment a good landmark has been the topic of research in various disciplines, including cognitive science, computer science, and urban planning. Lynch (1960) defines landmarks as physical entities that serve as external points of reference that stand out from their surroundings. Kaplan (1976) specified a landmark as “a known place for which the individual has a well-formed representation”. Although there are different definitions of landmarks, a common theme is that objects are considered landmarks if they have some kind of cognitive salience (both in terms of visual distinctiveness and frequency of interaction).

The usefulness of landmarks in route instructions has been shown in a number of different human-human studies. Experimental results from Lovelace et al. (1999) show that people not only use landmarks intuitively when giving directions, but they also perceive instructions that are given to them to be of higher quality when those instructions contain landmark information. Similar findings have also been reported by Michon and Denis (2001) and Tom and Denis (2003).

Regarding car navigation systems specifically, Burnett (2000) reports on a road-based user study which compared a landmark-based navigation system to a conventional car navigation system. Here the provision of landmark information in route directions led to a decrease of navigational errors. Furthermore, glances at the navigation display were shorter and fewer, which indicates less driver distraction in this particular experimental condition. Minimizing driver distraction

is a crucial goal of improved navigation systems, as driver inattention of various kinds is a leading cause of traffic accidents (25% of all police-reported car crashes in the US in 2000, according to Stutts et al. (2001)). Another road-based study conducted by May and Ross (2006) yielded similar results.

One recurring finding in studies on landmarks in navigation is that some user groups are able to benefit more from their inclusion than others. This is particularly the case for female users. While men tend to outperform women in wayfinding tasks, completing them faster and with fewer navigation errors (c.f. Allen (2000)), women are likely to show improved wayfinding performance when landmark information is given (e.g. Saucier et al. (2002)).

Despite all of this evidence from human-human studies, there has been remarkably little research on implemented navigation systems that use landmarks. Commercial systems make virtually no use of landmark information when giving directions, relying on metric representations instead (e.g. “Turn right in one hundred meters”). In academic research, there have only been a handful of relevant systems. A notable example is the DEEP MAP system, which was created in the SmartKom project as a mobile tourist information system for the city of Heidelberg (Malaka and Zipf, 2000; Malaka et al., 2004). DEEP MAP uses landmarks as waypoints for the planning of touristic routes for car drivers and pedestrians, while also making use of landmark information in the generation of route directions. Raubal and Winter (2002) combine data from digital city maps, facade images, cultural heritage information, and other sources to compute landmark descriptions that could be used in a pedestrian navigation system for the city of Vienna.

The key to the richness of these systems is a set of extensive, manually curated geographic and landmark databases. However, creation and maintenance of such databases is expensive, which makes it impractical to use these systems outside of the limited environments for which they were created. There have been a number of suggestions for automatically acquiring landmark data from existing electronic databases, for instance cadastral data (Elias, 2003) and airborne laser scans (Brenner and Elias, 2003). But the raw data for these approaches is still hard to obtain; informa-

tion about landmarks is mostly limited to geometric data and does not specify the semantic type of a landmark (such as “church”); and updating the landmark database frequently when the real world changes (e.g., a shop closes down) remains an open issue.

The closest system in the literature to the research we present here is the CORAL system (Dale et al., 2003). CORAL generates a text of driving instructions with landmarks out of the output of a commercial web-based route planner. Unlike CORAL, our system relies purely on open-source map data. Also, our system generates driving instructions in real time (as opposed to a single discourse before the user starts driving) and reacts in real time to driving errors. Finally, we evaluate our system thoroughly for driving errors, user satisfaction, and driver distraction on an actual driving task, and find a significant improvement over the baseline.

3 OpenStreetMap

A system that generates landmark-based route directions requires two kinds of data. First, it must plan routes between points in space, and therefore needs data on the road network, i.e. the road segments that make up streets along with their connections. Second, the system needs information about the landmarks that are present in the environment. This includes geographic information such as position, but also semantic information such as the landmark type.

We have argued above that the availability of such data has been a major bottleneck in the development of landmark-based navigation systems. In the Virtual Co-Pilot system, which we present below, we solve this problem by using data from OpenStreetMap, an on-line map resource that provides both types of information mentioned above, in a unified data structure. The OpenStreetMap project is to maps what Wikipedia is to encyclopedias: It is a map of the entire world which can be edited by anyone wishing to participate. New map data is usually added by volunteers who measure streets using GPS devices and annotate them via a Web interface. The decentralized nature of the data entry process means that when the world changes, the map will be updated quickly. Existing map data can be viewed as a zoomable map on the OpenStreetMap website, or it can be downloaded in an

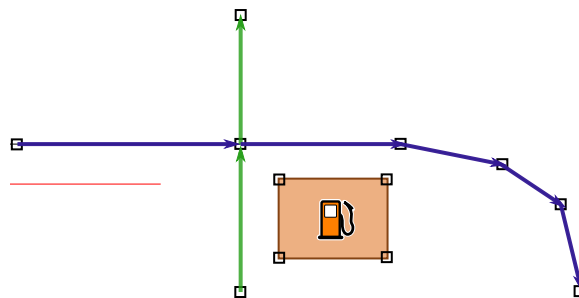


Figure 1: A graphical representation of some nodes and ways in OpenStreetMap.

	Landmark Type
Street Furniture	stop sign traffic lights pedestrian crossing
Visual Landmarks	church certain video stores certain supermarkets gas station pubs and bars

Figure 2: Landmarks used by the Virtual Co-Pilot.

XML format for offline use.

Geographical data in OpenStreetMap is represented in terms of *nodes* and *ways*. Nodes represent points in space, defined by their latitude and longitude. Ways consist of sequences of edges between adjacent nodes; we call the individual edges *segments* below. They are used to represent streets (with curved streets consisting of multiple straight segments approximating their shape), but also a variety of other real-world entities: buildings, rivers, trees, etc. Nodes and ways can both be enriched with further information by attaching *tags*. Tags encode a wide range of additional information using a predefined type ontology. Among other things, they specify the types of buildings (church, cafe, supermarket, etc.); where a shop or restaurant has a name, it too is specified in a tag. Fig. 1 is a graphical representation of some OpenStreetMap data, consisting of nodes and ways for two streets (with two and five segments) and a building which has been tagged as a gas station.

For the Virtual Co-Pilot system, we have chosen a set of concrete landmark types that we consider useful (Fig. 2). We operationalize the criteria for good landmarks sketched in Section 2 by requiring that a landmark should be easily *visible*, and that it should be *generic* in that it is appli-

cable not just for one particular city, but for any place for which OpenStreetMap data is available.

We end up with two classes of landmark types: *street furniture* and *visual landmarks*. Street furniture is a generic term for objects that are installed on streets. In this subset, we include *stop signs*, *traffic lights*, and *pedestrian crossings*. Our assumption is that these objects inherently possess a high salience, since they already require particular attention from the driver. “Visual landmarks” encompass roadside buildings that are not directly connected to the road infrastructure, but draw the driver’s attention due to visual salience. Churches are an obvious member of this group; in addition, we include gas stations, pubs, and bars, as well as certain supermarket and video store chains (selected for wide distribution over different cities and recognizable, colorful signs).

Given a certain location at which the Virtual Co-Pilot is to be used, we automatically extract suitable landmarks along with their types and locations from OpenStreetMap. We also gather the road network information that is required for route planning, and collect informations on streets, such as their names, from the tags. We then transform this information into a *directed street graph*. The nodes of this graph are the OpenStreetMap nodes that are part of streets; two adjacent nodes are connected by a single directed edge for segments of one-way streets and a directed edge in each direction for ordinary street segments. Each edge is weighted with the Euclidean distance between the two nodes.

4 Generation of route directions

We will now describe how the Virtual Co-Pilot generates route directions from OpenStreetMap data. The system generates three types of messages (see Fig. 3). First, at every *decision point*, i.e. at the intersection where a driving maneuver such as turning left or right is required, the user is told to turn immediately in the given direction (“now turn right”). Second, if the driver has followed an instruction correctly, we generate a *confirmation message* after the driver has made the turn, letting them know they are still on the right track. Finally, we generate *preview messages* on the street leading up to the decision point. These preview messages describe the location of the next driving maneuver.

Of the three types, preview messages are the

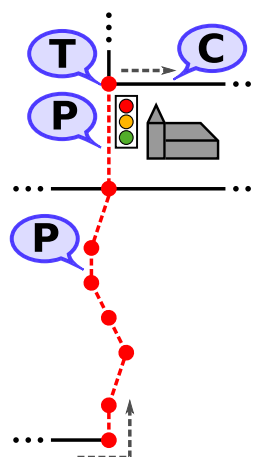


Figure 3: Schematic representation of an episode (dashed red line), with sample trigger positions of preview, turn instruction, and confirmation messages.

most interesting. Our system avoids the generation of metric distance indicators, as in “turn left in 100 meters”. Instead, it tries to find landmarks that describe the position of the decision point: “Prepare to turn left after the church.” When no landmark is available, the system tries to use street intersections as secondary landmarks, as in “Turn right at the next/second/third intersection.” Metric distances are only used when both of these strategies fail.

In-car NLG takes place in a heavily real-time setting, in which an utterance becomes uninterpretable or even misleading if it is given too late. This problem is exacerbated for NLG of speech because simply speaking the utterance takes time as well. One consequence that our system addresses is the problem of planning preview messages in such a way that they can be spoken before the decision point without overlapping each other. We handle this problem in the sentence planner, which may aggregate utterances to fit into the available time. A second problem is that the user’s reactions to the generated utterances are unpredictable; if the driver takes a wrong turn, the system must generate updated instructions in real time.

Below, we describe the individual components of the system. We mostly follow a standard NLG pipeline (Reiter and Dale, 2000), with a focus on the sentence planner and an extension to interactive real-time NLG.

Segment123
 From: Node1
 To: Node2
 On: "Main Street"

Segment124
 From: Node2
 To: Node3
 On: "Main Street"

Segment125
 From: Node3
 To: Node4
 On: "Park Street"

Segment126
 From: Node4
 To: Node5
 On: "Park Street"

Figure 4: A simple example of a route plan consisting of four street segments.

4.1 Content determination and text planning

The first step in our system is to obtain a plan for reaching the destination. To this end, we compute a shortest path on the directed street graph described in Section 3. The result is an ordered list of street segments that need to be traversed in the given order to successfully reach the destination; see Fig. 4 for an example.

To be suitable as the input for an NLG system, this flat list of OpenStreetMap nodes needs to be subdivided into smaller message chunks. In turn-by-turn navigation, the general delimiter between such chunks are the driving maneuvers that the driver must execute at each decision point. We call each span between two decision points an *episode*. Episodes are not explicitly represented in the original route plan: although every segment has a street name associated with it, the name of a street sometimes changes as we go along, and because chains of segments are used to model curved streets in OpenStreetMap, even segments that are joined at an angle may be parts of the same street. Thus, in Fig. 4 it is not apparent which segment traversals require any navigational maneuvers.

We identify episode boundaries with the following heuristic. We first assume that episode boundaries occur when the street name changes from one segment to the next. However, staying on the road may involve a driving maneuver (and therefore a decision point) as well, e.g.

when the road makes a sharp turn where a minor street forks off. To handle this case, we introduce decision points at nodes with multiple adjacent segments if the angle between the incoming and outgoing segment of the street exceeds a certain threshold. Conversely, our heuristic will sometimes end an episode where no driving maneuver is necessary, e.g. when an ongoing street changes its name. This is unproblematic in practice; the system will simply generate an instruction to keep driving straight ahead. Fig. 3 shows a graphical representation of an episode, with the street segments belonging to it drawn as red dashed lines.

4.2 Aggregation

Because we generate spoken instructions that are given to the user while they are driving, the timing of the instructions becomes a crucial issue, especially because a driver moves faster than the user of a pedestrian navigation system. It is undesirable for a second instruction to interrupt an earlier one. On the other hand, the second instruction cannot be delayed because this might make the user miss a turn or interpret the instruction incorrectly.

We must therefore control at which points instructions are given and make sure that they do not overlap. We do this by always presenting preview messages at *trigger positions* at certain fixed distances from the decision point. The sentence planner calculates where these trigger positions are located for each episode. In this way, we create time frames during which there is enough time for instructions to be presented.

However, some episodes are too short to accommodate the three trigger positions for the confirmation message and the two preview messages. In such episodes, we aggregate different messages. We remove the trigger positions for the two preview messages from the episode, and instead add the first preview message to the turn instruction message of the previous episode. This allows our system to generate instructions like "Now turn right, and then turn left after the church."

4.3 Generation of landmark descriptions

The Virtual Co-Pilot computes referring expressions to decision points by selecting appropriate landmarks. To this end, it first looks up landmark candidates within a given range of the decision point from the database created in Section 3. This

yields an initial list of landmark candidates.

Some of these landmark candidates may be unsuitable for the given situation because of lack of *uniqueness*. If there are several visual landmarks of the same type along the course of an episode, all of these landmark candidates are removed. For episodes which contain multiple street furniture landmarks of the same type, the first three in each episode are retained; a referring expression for the decision point might then be “at the second traffic light”. If the decision point is no more than three intersections away, we also add a landmark description of the form “at the third intersection”. Furthermore, a landmark must be visible from the last segment of the current episode; we only retain a candidate if it is either adjacent to a segment of the current episode or if it is close to the end point of the very last segment of the episode. Among the landmarks that are left over, the system prefers visual landmarks over street furniture, and street furniture over intersections. If no landmark candidates are left over, the system falls back to metric distances.

Second, the Virtual Co-Pilot determines the spatial relationship between the landmark and the decision point so that an appropriate preposition can be used in the referring expression. If the decision point occurs before the landmark along the course of the episode, we use the preposition “in front of”, otherwise, we use “after”. Intersections are always used with “at” and metric distances with “in”.

Finally, the system decides how to refer to the landmark objects themselves. Although it has access to the names of all objects from the OpenStreetMap data, the user may not know these names. We therefore refer to churches, gas stations, and any street furniture simply as “the church”, “the gas station”, etc. For supermarkets and bars, we assume that these buildings are more saliently referred to by their names, which are used in everyday language, and therefore use the names to refer to them.

The result of the sentence planning stage is a list of semantic representations, specifying the individual instructions that are to be uttered in each episode; an example is shown in Fig. 5. For each type of instruction, we then use a sentence template to generate linguistic surface forms by inserting the information contained in those plans into the slots provided by the templates (e.g.

Preview message p_1 :
Trigger position: Node3 – 50m
Turn direction: right
Landmark: church
Preposition: after

Preview message $p_2 = p_1$, except:
Trigger position: Node3 – 100m

Turn instruction t_1 :
Trigger position: Node3
Turn direction: right

Confirmation message c_1 :
Trigger position: Node3 + 50m

Figure 5: Semantic representations of the different types of instructions in one episode.

“Turn *direction preposition landmark*”).

4.4 Interactive generation

As a final point, the NLG process of a car navigation system takes place in an *interactive* setting: as the system generates and utters instructions, the user may either follow them correctly, or they may miss a turn or turn incorrectly because they misunderstood the instruction or were forced to disregard it by the traffic situation. The system must be able to detect such problems, recover from them, and generate new instructions in real time.

Our system receives a continuous stream of information about the position and direction of the user. It performs *execution monitoring* to check whether the user is still following the intended route. If a trigger position is reached, we present the instruction that we have generated for this position. If the user has left the route, the system reacts by planning a new route starting from the user’s current position and generating a new set of instructions. We check whether the user is following the intended route in the following way. The system keeps track of the current episode of the route plan, and monitors the distance of the car to the final node of the episode. While the user is following the route correctly, the distance between the car and the final node should decrease or at least stay the same between two measurements. To accommodate for occasional deviations from the middle of the road, we allow five subsequent measurements to increase the distance; the sixth increase of the distance triggers a recomputation of the route plan and a freshly generated instruction. On the other hand, when the distance

of the car to the final node falls below a certain threshold, we assume that the end of the episode has been reached, and activate the next episode. By monitoring whether the user is now approaching the final node of this new episode, we can in particular detect wrong turns at intersections.

Because each instruction carries the risk that it may not be followed correctly, there is a question as to whether it is worth planning out all remaining instructions for the complete route plan. After all, if the user does not follow the first instruction, the computation of all remaining instructions was a waste of time. We decided to compute all future instructions anyway because the aggregation procedure described above requires them. In practice, the NLG process is so efficient that all instructions can be done in real time, but this decision would have to be revisited for a slower system.

5 Evaluation

We will now report on an experiment in which we evaluated the performance of the Virtual Co-Pilot.

5.1 Experimental Method

5.1.1 Subjects

In total, 12 participants were recruited through printed ads and mailing lists. All of them were university students aged between 21 and 27 years. Our experiment was balanced for gender, hence we recruited 6 male and 6 female participants. All participants were compensated for their effort.

5.1.2 Design

The driving simulator used in the experiment replicates a real-world city center using a 3D model that contains buildings and streets as they can be perceived in reality. The street layout 3D model used by the driving simulator is based on OpenStreetMap data, and buildings were added to the virtual environment based on cadastral data. To increase the perceived realism of the model, some buildings were manually enhanced with photographic images of their real-world counterparts (see Fig. 7).

Figure 6 shows the set-up of the evaluation experiment. The virtual driving simulator environment (main picture in Fig. 7) was presented to the participants on a 20" computer screen (A). In addition, graphical navigation instructions (shown in the lower right of Fig. 7) were displayed on

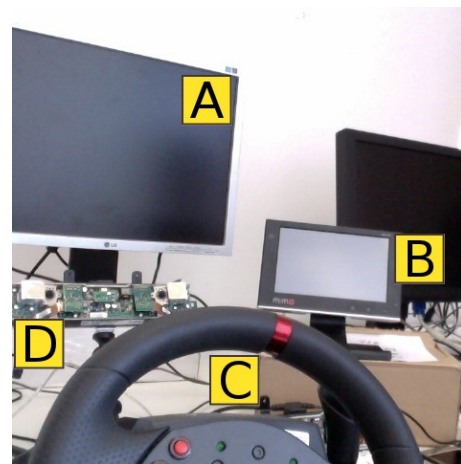


Figure 6: Experiment setup. A) Main screen B) Navigation screen C) steering wheel D) eye tracker

a separate 7" monitor (B). The driving simulator was controlled by means of a steering wheel (C), along with a pair of brake and acceleration pedals. We recorded user eye movements using a Tobii IS-Z1 table-mounted eye tracker (D). The generated instructions were converted to speech using MARY, an open-source text-to-speech system (Schröder and Trouvain, 2003), and played back on loudspeakers.

The task of the user was to drive the car in the virtual environment towards a given destination; spoken instructions were presented to them as they were driving, in real time. Using the steering wheel and the pedals, users had full control over steering angles, acceleration and braking. The driving speed was limited to 30 km/h, but there were no restrictions otherwise. The driving simulator sent the NLG system a message with the current position of the car (as GPS coordinates) once per second.

Each user was asked to drive three short routes in the driving simulator. Each route took about four minutes to complete, and the travelled distance was about 1 km. The number of episodes per route ranged from three to five. Landmark candidates were sufficiently dense that the Virtual Co-Pilot used landmarks to refer to all decision points and never had to fall back to the metric distance strategy.

There were three experimental conditions, which differed with respect to the spoken route instructions and the use of the navigation screen. In the baseline condition, designed to replicate the behavior of an off-the-shelf commercial car nav-

	All Users		Males		Females	
	B	VCP	B	VCP	B	VCP
Total Fixation Duration (seconds)	4.9	3.5	2.7	4.1	7.0	2.9*
Total Fixation Count (N)	21.8	15.4	13.5	16.5	30.0	14.3*
"The system provided the right amount of information at any time"	3.9	2.9	4.2*	3.3	3.5	2.5
"I was insecure at times about still being on the right track."	2.3	3.2	1.9*	2.8	2.6	3.5
"It was important to have a visual representation of route directions"	4.3	4.0	4.2	4.2	4.3	3.7
"I could trust the navigation system"	3.6	3.7	4.1	3.7	3.0	3.7

Figure 8: Mean values for gaze behavior and subjective evaluation, separated by user group and condition (B = baseline, VCP = our system). Significant differences are indicated by *; better values are printed in boldface.



Figure 7: Screenshot of a scene in the driving simulator. Lower right corner: matching screenshot of navigation display.

igation system, participants were provided with spoken metric distance-to-turn navigation instructions. The navigation screen showed arrows depicting the direction of the next turn, along with the distance to the decision point (cf. Fig. 7). The second condition replaced the spoken route instructions by those generated by the Virtual Co-Pilot. In a third condition, the output of the navigation screen was further changed to display an icon for the next landmark along with the arrow and distance indicator. The three routes were presented to the users in different orders, and combined with the conditions in a Latin Squares design. In this paper, we focus on the first and second condition, in order to contrast the two styles of spoken instruction.

Participants were asked to answer two questionnaires after each trial run. The first was the DALI questionnaire (Pauzié, 2008), which asks subjects to report how they perceived different

aspects of their cognitive workload (general, visual, auditive and temporal workload, as well as perceived stress level). In the second questionnaire, participants were state to rate their agreement with a number of statements about their subjective impression of the system on a 5-point unlabelled Likert scale, e.g. whether they had received instructions at the right time or whether they trusted the navigation system to give them the right instructions during trials.

5.2 Results

There were no significant differences between the Virtual Co-Pilot and the baseline system on task completion time, rate of driving errors, or any of the questions of the DALI questionnaire. Driving errors in particular were very rare: there were only four driving errors in total, two of which were due to problems with left/right coordination.

We then analyzed the gaze data collected by the table-mounted eye tracker, which we set up such that it recognized glances at the navigation screen. In particular, we looked at the *total fixation duration* (TFD), i.e. the total amount of time that a user spent looking at the navigation screen during a given trial run. We also looked at the *total fixation count* (TFC), i.e. the total number of times that a user looked at the navigation screen in each run. Mean values for both metrics are given in Fig. 8, averaged over all subjects and only male and female subjects, respectively; the "VCP" column is for the Virtual Co-Pilot, whereas "B" stands for the baseline. We found that male users tended to look more at the navigation screen in the VCP condition than in B, although the difference is not statistically significant. However, female users looked at the navigation screen significantly fewer

times ($t(5) = 3.2, p < 0.05$, t-test for dependent samples) and for significantly shorter amounts of time ($t(5) = 3.2, p < 0.05$) in the VCP condition than in B.

On the subjective questionnaire, most questions yielded no significant differences (and are not reported here). However, we found that female users tended to rate the Virtual Co-Pilot more positively than the baseline on questions concerning trust in the system and the need for the navigation screen (but not significantly). Male users found that the baseline significantly outperformed the Virtual Co-Pilot on presenting instructions at the right time ($t(5) = 2.7, p < 0.05$) and on giving them a sense of security in still being on the right track ($t(5) = -2.7, p < 0.05$).

5.3 Discussion

The most striking result of the evaluation is that there was a significant reduction of looks to the navigation display, even if only for one group of users. Female users looked at the navigation screen less and more rarely with the Virtual Co-Pilot compared to the baseline system. In a real car navigation system, this translates into a driver who spends less time looking away from the road, i.e. a reduction in driver distraction and an increase in traffic safety. This suggests that female users learned to trust the landmark-based instructions, an interpretation that is further supported by the trends we found in the subjective questionnaire.

We did not find these differences in the male user group. Part of the reason may be the known gender differences in landmark use we mentioned in Section 2. But interestingly, the two significantly worse ratings by male users concerned the correct timing of instructions and the feedback for driving errors, i.e. issues regarding the system's real-time capabilities. Although our system does not yet perform ideally on these measures, this confirms our initial hypothesis that the NLG system must track the user's behavior and schedule its utterances appropriately. This means that earlier systems such as CORAL, which only compute a one-shot discourse of route instructions without regard to the timing of the presentation, miss a crucial part of the problem.

Apart from the exceptions we just discussed, the landmark-based system tended to score comparably or a bit worse than the baseline on the

other subjective questions. This may partly be due to the fact that the subjects were familiar with existing commercial car navigation systems and not used to landmark-based instructions. On the other hand, this finding is also consistent with results of other evaluations of NLG systems, in which an improvement in the objective task usefulness of the system does not necessarily correlate with improved scores from subjective questionnaires (Gatt et al., 2009).

6 Conclusion

In this paper, we have described a system for generating real-time car navigation instructions with landmarks. Our system is distinguished from earlier work in its reliance on open-source map data from OpenStreetMap, from which we extract both the street graph and the potential landmarks. This demonstrates that open resources are now informative enough for use in wide-coverage navigation NLG systems. The system then chooses appropriate landmarks at decision points, and continuously monitors the driver's behavior to provide modified instructions in real time when driving errors occur.

We evaluated our system using a driving simulator with respect to driving errors, user satisfaction, and driver distraction. To our knowledge, we have shown for the first time that a landmark-based car navigation system outperforms a baseline significantly; namely, in the amount of time female users spend looking away from the road.

In many ways, the Virtual Co-Pilot is a very simple system, which we see primarily as a starting point for future research. The evaluation confirmed the importance of interactive real-time NLG for navigation, and we therefore see this as a key direction of future work. On the other hand, it would be desirable to generate more complex referring expressions ("the tall church"). This would require more informative map data, as well as a formal model of visual salience (Kelleher and van Genabith, 2004; Raubal and Winter, 2002).

Acknowledgments. We would like to thank the DFKI CARMINA group for providing the driving simulator, as well as their support. We would furthermore like to thank the DFKI Agents and Simulated Reality group for providing the 3D city model.

References

- G. L. Allen. 2000. Principles and practices for communicating route knowledge. *Applied Cognitive Psychology*, 14(4):333–359.
- C. Brenner and B. Elias. 2003. Extracting landmarks for car navigation systems using existing gis databases and laser scanning. *International archives of photogrammetry remote sensing and spatial information sciences*, 34(3/W8):131–138.
- G. Burnett. 2000. ‘Turn right at the Traffic Lights’: The Requirement for Landmarks in Vehicle Navigation Systems. *The Journal of Navigation*, 53(03):499–510.
- R. Dale, S. Geldof, and J. P. Prost. 2003. Using natural language generation for navigational assistance. In *ACSC*, pages 35–44.
- B. Elias. 2003. Extracting landmarks with data mining methods. *Spatial information theory*, pages 375–389.
- A. Gatt, F. Portet, E. Reiter, J. Hunter, S. Mahamood, W. Moncur, and S. Sripatha. 2009. From data to text in the neonatal intensive care unit: Using NLG technology for decision support and information management. *AI Communications*, 22:153–186.
- S. Kaplan. 1976. Adaption, structure and knowledge. In G. Moore and R. Gollidge, editors, *Environmental knowing: Theories, research and methods*, pages 32–45. Dowden, Hutchinson and Ross.
- J. D. Kelleher and J. van Genabith. 2004. Visual salience and reference resolution in simulated 3-D environments. *Artificial Intelligence Review*, 21(3).
- A. Koller, K. Striegnitz, D. Byron, J. Cassell, R. Dale, J. Moore, and J. Oberlander. 2010. The First Challenge on Generating Instructions in Virtual Environments. In E. Krahmer and M. Theune, editors, *Empirical Methods in Natural Language Generation*. Springer.
- N. Lessmann, S. Kopp, and I. Wachsmuth. 2006. Situated interaction with a virtual human – perception, action, and cognition. In G. Rickheit and I. Wachsmuth, editors, *Situated Communication*, pages 287–323. Mouton de Gruyter.
- K. Lovelace, M. Hegarty, and D. Montello. 1999. Elements of good route directions in familiar and unfamiliar environments. *Spatial information theory. Cognitive and computational foundations of geographic information science*, pages 751–751.
- K. Lynch. 1960. *The image of the city*. MIT Press.
- R. Malaka and A. Zipf. 2000. DEEP MAP – Challenging IT research in the framework of a tourist information system. *Information and communication technologies in tourism*, 7:15–27.
- R. Malaka, J. Haeussler, and H. Aras. 2004. SmartKom mobile: intelligent ubiquitous user interaction. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*.
- A. J. May and T. Ross. 2006. Presence and quality of navigational landmarks: effect on driver performance and implications for design. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(2):346.
- P. E. Michon and M. Denis. 2001. When and why are visual landmarks used in giving directions? *Spatial information theory*, pages 292–305.
- A. Pauzié. 2008. Evaluating driver mental workload using the driving activity load index (DALI). In *Proc. of European Conference on Human Interface Design for Intelligent Transport Systems*, pages 67–77.
- M. Raubal and S. Winter. 2002. Enriching wayfinding instructions with local landmarks. *Geographic information science*, pages 243–259.
- E. Reiter and R. Dale. 2000. *Building natural language generation systems*. Studies in natural language processing. Cambridge University Press.
- D. M. Saucier, S. M. Green, J. Leason, A. MacFadden, S. Bell, and L. J. Elias. 2002. Are sex differences in navigation caused by sexually dimorphic strategies or by differences in the ability to use the strategies?. *Behavioral Neuroscience*, 116(3):403.
- M. Schröder and J. Trouvain. 2003. The German text-to-speech synthesis system MARY: A tool for research, development and teaching. *International Journal of Speech Technology*, 6(4):365–377.
- K. Striegnitz and F. Majda. 2009. Landmarks in navigation instructions for a virtual environment. *Online Proceedings of the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE-1)*.
- J. C. Stutts, D. W. Reinfurt, L. Staplin, and E. A. Rodgman. 2001. The role of driver distraction in traffic crashes. *Washington, DC: AAA Foundation for Traffic Safety*.
- A. Tom and M. Denis. 2003. Referring to landmark or street information in route directions: What difference does it make? *Spatial information theory*, pages 362–374.