A NLG-based Application for Walking Directions

Michael Roth and Anette Frank
Department of Computational Linguistics
Heidelberg University
69120 Heidelberg, Germany
{mroth,frank}@cl.uni-heidelberg.de

Abstract

This work describes an online application that uses Natural Language Generation (NLG) methods to generate walking directions in combination with dynamic 2D visualisation. We make use of third party resources, which provide for a given query (geographic) routes and landmarks along the way. We present a statistical model that can be used for generating natural language directions. This model is trained on a corpus of walking directions annotated with POS, grammatical information, frame-semantics and markup for temporal structure.

1 Introduction

The purpose of route directions is to inform a person, who is typically not familiar with his current environment, of how to get to a designated goal. Generating such directions poses difficulties on various conceptual levels such as the planning of the route, the selection of landmarks along the way (i.e. easily recognizable buildings or structures) and generating the actual instructions of how to navigate along the route using the selected landmarks as reference points.

As pointed out by Tom & Denis (2003), the use of landmarks in route directions allows for more effective way-finding than directions relying solely on street names and distance measures. An experiment performed in Tom & Denis' work also showed that people tend to use landmarks rather than street names when producing route directions themselves.

The application presented here is an early research prototype that takes a data-driven generation approach, making use of annotated corpora collected in a way-finding study. In contrast to previously developed NLG systems in this area (e.g. Dale et. al, 2002), one of our key features is the integration of a number of online resources to compute routes and to find salient landmarks. The information acquired from these resources can then be used to generate natural directions that are both easier to memorise and easier to follow than directions given by a classic route planner or navigation system.

The remainder of this paper is structured as follows: In Section 2 we introduce our system and describe the resources and their integration in the architecture. Section 3 describes our corpus-based generation approach, with Section 4 outlining our integration of text generation and visualisation. Finally, Section 5 gives a short conclusion and discusses future work.

2 Combining Resources

The route planner used in our system is provided by the *Google Maps API*[1]. Given a route computed in Google Maps, our system queries a number of online resources to determine landmarks that are adjacent to this route. At the time of writing, these resources are: *OpenStreetMaps*[2] for public transportation, the Wikipedia *WikiProject Geographical coordinates*[3] for salient buildings, statues and other objects, *Google AJAX Search API*[4] for "yellow pages landmarks" such as hotels and restaurants, and *Wikimapia*[5] for squares and other prominent places.

All of the above mentioned resources can be queried for landmarks either by a single GPS

[1] http://code.google.com/apis/maps/
[2] http://www.openstreetmap.org
[3] http://en.wikipedia.org/wiki/Wikipedia:WikiProject Geographical_coordinates
[4] http://code.google.com/apis/ajaxsearch
[5] http://www.wikimapia.org

Proceedings of the ACL-IJCNLP 2009 Software Demonstrations, pages 37–40,
Suntec, Singapore, 3 August 2009.

coordinate (using the `LocalSearch` method in Google AJAX Search and web tools in Wikipedia) or an area of GPS coordinates (using URL based queries in Wikimapia and OpenStreetMaps). The following list describes the data formats returned by the respective services and how they were integrated:

- **Wikimapia** and **OpenStreetMaps** – Both resources return landmarks in the queried area as an XML file that specifies GPS coordinates and additional information. The XML files are parsed using a JavaScript implementation of a SAX parser. The coordinates and names of landmarks are then used to add objects within the Google Maps API.
- **Wikipedia** – In order to integrate landmarks from Wikipedia, we make use of a community created tool called *search-a-place* [6], which returns landmarks from Wikipedia in a given radius of a GPS coordinate. The results are returned in an HTML table that is converted to an XML file similar to the output of Wikimapia. Both the query and the conversion are implemented in a *Yahoo! Pipe*[7] that can be accessed in JavaScript via its URL.
- **Google AJAX Search** – The results returned by the Google AJAX Search API are JavaScript objects that can be directly inserted in the visualisation using the Google Maps API.

3 Using Corpora for Generation

A data-driven generation approach achieves a number of advantages over traditional approaches for our scenario. First of all, corpus data can be used to learn directly how certain events are typically expressed in natural language, thus avoiding the need of manually specifying linguistic realisations. Secondly, variations of discourse structures found in naturally given directions can be learned and reproduced to avoid monotonous descriptions in the generation part. Last but not least, a corpus with good coverage can help us determine the correct selection restrictions on verbs and nouns occurring in directions. The price to pay for these advantages is the cost of annotation; however we believe that this is a reasonable trade-off, in view of the fact that a small annotated corpus and reasonable generalizations in data modelling will likely yield enough information for the intended navigation applications.

3.1 Data Collection

We currently use the data set from (Marciniak & Strube, 2005) to learn linguistic expressions for our generation approach. The data is annotated on the following levels:

- Token and POS level
- Grammatical level (including annotations of main verbs, arguments and connectives)
- Frame-semantics level (including semantic roles and frame annotations in the sense of (Fillmore, 1977))
- Temporal level (including temporal relations between discourse units)

3.2 Our Generation Approach

At the time of writing, our system only makes use of the first three annotation levels. The lexical selection is inspired by the work of Ratnaparkhi (2000) with the overall process designed as follows: given a certain situation on a route, our generation component receives the respective frame name and a list of semantic role filling landmarks as input (cf. Section 4). The generation component then determines a list of potential lexical items to express this frame using the relative frequencies of verbs annotated as evoking the particular frame with the respective set of semantic roles (examples in Table 1).

SELF_MOTION	
PATH	17% *walk*, 13% *follow*, 10% *cross*, 7% *continue*, 6% *take*, …
GOAL	18% *get*, 18% *enter*, 9% *continue*, 7% *head*, 5% *reach*, …
SOURCE	14% *leave*, 14% *start*, …
DIRECTION	25% *continue*, 13% *make*, 13% *walk*, 6% *go*, 3% *take*, …
DISTANCE	15% *continue*, 8% *go*, ...
PATH + GOAL	29% *continue*, 14% *take*, ...
DISTANCE + GOAL	100% *walk*
DIRECTION + PATH	23% *continue*, 23% *walk*, 8% *take*, 6% *turn*, 6% *face*, …

Table 1: Probabilities of lexical items for the frame SELF_MOTION and different frame elements

[6] http://toolserver.org/~kolossos/wp-world/umkreis.php

[7] http://pipes.yahoo.com/pipes/pipe.info?_id=BBI0x8G73RGbWzKnBR50VA

For frame-evoking elements and each associated semantic role-filler in the situation, the grammatical knowledge learned from the annotation level determines how these parts can be put together in order to generate a full sentence (cf. Table 2).

SELF_MOTION	
walk + [*building* $_{\text{PATH}}$]	*walk* → *walk* + PP PP → along + NP NP → the + *building*
get + [*building* $_{\text{GOAL}}$]	*get* → *get* + to + NP NP → the + *building*
take + [*left* $_{\text{DIRECTION}}$]	*take* → *take* + NP NP → a + *left*

Table 2: Examples of phrase structures for the frame SELF_MOTION and different semantic role fillers

4 Combining Text and Visualisation

As mentioned in the previous section, our model is able to compute single instructions at crucial points of a route. At the time of writing the actual integration of this component consists of a set of hardcoded rules that map route segments to frames, and landmarks within the segment to role fillers of the considered frame. The rules are specified as follows:

- A turning point given by the Google Maps API is mapped to the SELF_MOTION frame with the actual direction as the semantic role *direction*. If there is a landmark adjacent to the turning point, it is added to the frame as the role filler of the role *source*.
- If a landmark is adjacent or within the starting point of the route, it will be mapped to the SELF_MOTION frame with the landmark filling the semantic role *source*.
- If a landmark is adjacent or within the goal of a route, it will be mapped to the SELF_MOTION frame with the landmark filling the semantic role *goal*.
- If a landmark is adjacent to a route or a route segment is within a landmark, the respective segment will be mapped to the SELF_MOTION frame with the landmark filling the semantic role *path*.

5 Conclusions and Outlook

We have presented the technical details of an early research prototype that uses NLG methods to generate walking directions for routes computed by an online route planner. We outlined the advantages of a data-driven generation approach over traditional rule-based approaches and implemented a first-version application, which can be used as an initial prototype extensible for further research and development.

Our next goal in developing this system is to enhance the generation component with an integrated model based on machine learning techniques that will also account for discourse level phenomena typically found in natural language directions. We further intend to replace the current hard-coded set of mapping rules with an automatically induced mapping that aligns physical routes and landmarks with the semantic representations. The application is planned to be used in web experiments to acquire further data for alignment and to study specific effects in the generation of walking instructions in a multimodal setting.

The prototype system described above will be made publicly available at the time of publication.

Acknowledgements

This work is supported by the DFG-financed innovation fund FRONTIER as part of the Excellence Initiative at Heidelberg University (ZUK 49/1).

References

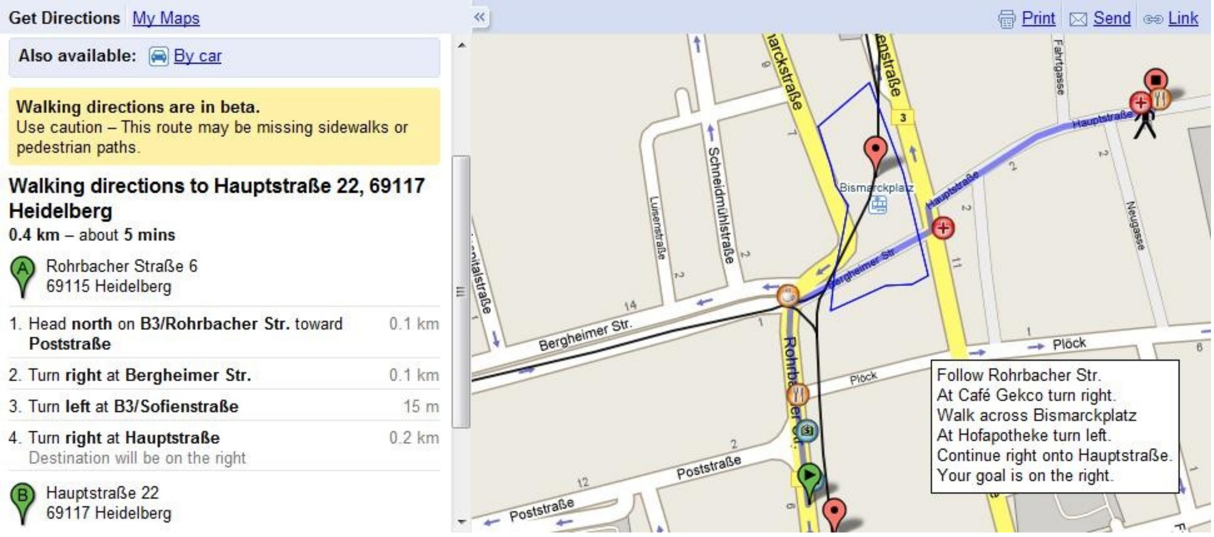
Dale, R., Geldof, S., & Prost, J.-P. (2002). Generating more natural route descriptions. *Proceedings of the 2002 Australasian Natural Language Processing Workshop.* Canberra, Australia.

Fillmore, C. (1977). The need for a frame semantics in linguistics. *Methods in Linguistics , 12*, 2-29.

Marciniak, T., & Strube, M. (2005). Using an annotated corpus as a knowledge source for language generation. *Proceedings of the Workshop on Using Corpora for Natural Language Generation*, (pp. 19-24). Birmingham, UK.

Ratnaparkhi, A. (2000). Trainable Methods for Surface Natural Language Generation. *Proceedings of the 6th Applied Natural Language Processing Conference.* Seattle, WA, USA.

Tom, A., & Denis, M. (2003). Referring to landmark or street information in route directions: What difference does it make? In W. Kuhn, M. Worboys, & S. Timpf (Eds.), *Spatial Information Theory* (pp. 384-397). Berlin: Springer.



Get Directions My Maps

Also available: By car

Walking directions are in beta.
Use caution – This route may be missing sidewalks or pedestrian paths.

Walking directions to Hauptstraße 22, 69117 Heidelberg
0.4 km – about 5 mins

A Rohrbacher Straße 6
69115 Heidelberg

1. Head **north** on **B3/Rohrbacher Str.** toward **Poststraße** 0.1 km
2. Turn **right** at **Bergheimer Str.** 0.1 km
3. Turn **left** at **B3/Sofienstraße** 15 m
4. Turn **right** at **Hauptstraße** 0.2 km
Destination will be on the right

B Hauptstraße 22
69117 Heidelberg

Follow Rohrbacher Str.
At Café Gekco turn right.
Walk across Bismarckplatz
At Hofapotheke turn left.
Continue right onto Hauptstraße.
Your goal is on the right.

Figure 1: Visualised route from *Rohrbacher Straße 6* to *Hauptstrasse 22, Heidelberg*. Left: GoogleMaps directions; Right: GoogleMaps visualisation enriched with landmarks and directions generated by our system (The directions were manually inserted here as they are actually presented step-by-step following the route)

Script Outline

Our demonstration is outlined as follows: At first we will have a look at the textual outputs of standard route planners and discuss at which points the respective instructions could be improved in order to be better understandable or easier to follow. We will then give an overview of different types of landmarks and argue how their integration into route directions is a valuable step towards better and more natural instructions.

Following the motivation of our work, we will present different online resources that provide landmarks of various sorts. We will look at the information provided by these resources, examine the respective input and output formats, and state how the formats are integrated into a common data representation in order to access the information within the presented application.

Next, we will give a brief overview of the corpus in use and point out which kinds of annotations were available to train the statistical generation component. We will discuss which other annotation levels would be useful in this scenario and which disadvantages we see in the current corpus. Subsequently we outline our plans to acquire further data by collecting directions for routes computed via Google Maps, which would allow an easier alignment between the instructions and routes.

Finally, we conclude the demonstration with a presentation of our system in action. During the presentation, the audience will be given the possibility to ask questions and propose routes for which we show our system's computation and output (cf. Figure 1).

System Requirements

The system is currently developed as a web-based application that can be viewed with any JavaScript supporting browser. A mid-end CPU is required to view the dynamic route presentation given by the application. Depending on the presentation mode, we can bring our own laptop so that the only requirements to the local organisers would be a stable internet connection (access to the resources mentioned in the system description is required) and presentation hardware (projector or sufficiently large display).