Knowledge-Based Target Selection in a Building Navigation System

Wojciech Palacz
The Faculty of Physics, Astronomy and Applied Computer Science,
Jagiellonian University, Kraków, Poland
wojciech.palacz@uj.edu.pl

Abstract. This paper is concerned with indoor navigation systems integrated with knowledge bases containing rich semantic data about buildings and their constituent spaces. Specifically, it is concerned with a system which utilizes graph models with labels, attributes and hierarchies storing semantic knowledge. That information can be used to describe navigation targets, as was shown in (Palacz et al., 2016). This paper continues that work and proposes a graphical user interface which allows for specifying complex target selection criteria.

1. Introduction

This paper is concerned with indoor pedestrian navigation systems. Their most obvious application is the case of a first-time visitor, who needs to get to a specific room in an unknown building while knowing only the number of this room. To help him in this task a navigation system needs a model of the building which allows for finding the optimal path between any two points, a way of determining the point where the visitor is currently standing, and a database containing numbers assigned to rooms in the building.

A room number is an example of a piece of semantic data attached to a particular space in a building. There are many more types of semantic data which are associated with buildings and their constituent areas. For example, knowledge about names of room occupants in an office building, about different room types in a university building (office room, lecture hall, laboratory, canteen, etc.), or about organizational units and which rooms are assigned to them.

Having this knowledge in its database makes the navigation system useful to people who already know the building and its room numbering scheme, but require a bit of help with finding targets described in terms of this additional semantic data. (Palacz et al., 2016) contains examples of such target descriptions. Some of them correspond to a single point/room (for example, “the coffee machine closest to my current position” or “the closest laboratory belonging to the Department of Organic Chemistry”), other to a set of points (“all office rooms belonging to the Department of Games”).

The system proposed in (Palacz et al., 2016) uses graph-based data structures. Graphs are an obvious choice for any task related to path planning. All navigation systems use graphs with nodes representing places and edges representing costs of moving between them. This “cost” can mean a distance measured in kilometers, time required, expended liters of gasoline, or something else—this depends on what the navigation system is designed to optimize for. In our case, edges are labelled with time required by the average human to walk from one endpoint to the other.

Additional attributes and hierarchies are used to store semantic knowledge. For example, the models have hierarchies assigning nodes representing rooms as children of nodes representing functional areas of the building, as well as appropriate administrative departments. They also have attributes storing information about door labels (“room no 108”), about presence of specific equipment (e.g. big-format printers), and so on.

As an example of advanced target selection, (Palacz et al., 2016) discussed an algorithm which directs the user to rooms represented by graph nodes with a specific attribute, visiting them in the order determined by their position in the administrative hierarchy. That algorithm had to be implemented in the navigation application's code before the users could take advantage of it. While that approach is suitable for systems which support a few well-specified tasks, it does not scale and cannot be used in a general, multipurpose navigation system.

As the size and richness of the knowledge base grows, so does exponentially grow the number of target selection algorithms which can be envisioned. Implementing all of them in advance, on the remote chance that some user in the future will have a need for a particular one, is unfeasible. Therefore, this paper proposes extending the navigation application with a target selection GUI displaying available selection criteria. It will allow the user to choose those applicable to his current task, to combine them using boolean operators, and to arrange them into successive phases.

In order to be tightly aligned with the knowledge base the GUI has to be dynamically generated. Let us consider a building in which all rooms are assigned to the same department—in this case, including "administrative unit" among the selection criteria is unnecessary, maybe even confusing. It is also necessary to note that the graph model is extensible and may be modified rapidly. The navigation application has to be able to adapt its GUI to changes like updated labels or introduction of a new graph attribute.

2. Indoor Navigation Systems

The field of indoor navigation includes many research problems. The most obvious one is determining the current position of the user of a navigation system. In contrast to the outdoor systems, an indoor system cannot rely on satellite-based location methods, because they are unreliable inside buildings (Gilliéron et al., 2013).

Therefore, indoor systems usually determine their position by taking a fix on stationary devices installed in the building. Wi-Fi access points are often used in this role, because they are already present in nearly all contemporary buildings. An example of such a Wi-Fi-based system deployed on a university campus can be found in (Köbben et al., 2006).

Wi-Fi-based methods have rather low accuracy. In cases where more precision is required, a network of dedicated reference devices must be deployed. This network can use RFID tags, Bluetooth beacons, etc. The Meridian Platform (Aruba Networks, 2019) is a representative example of a commercially available, Bluetooth-based solution. It requires beacons which are placed no more than 10 meters apart and provides accuracy of about 4 meters.

A systematic review of different methods and sensors used for determining the user's location can be found in, e.g., (Pahlavan et al., 2002) or (Correa et al., 2017).

Another research problem concerns navigation maps. A navigation system needs a map of the building, and it needs to be some kind of a graph with nodes representing specific points in the building and edges representing available paths between these points. This map can be constructed manually by developers of the system, but this is time-consuming and can introduce errors.

Contemporary buildings are constructed using the BIM (Building Information Modeling) methodology. Thus, a computer-readable BIM model of the building must be available, often in the form of an IFC file (buildingSMART International, 2019). Using this model as a basis for navigation map is a much more attractive solution.

A BIM model can be compared to an object-oriented database. It is composed of IFC entities arranged in a hierarchical manner. Each entity includes a fixed number of attributes, defined in the IFC standard, and an arbitrary number of additional properties. There are three fundamental entity types: IfcObjectDefinition, IfcPropertyDefinition and IfcRelationship. They in turn have

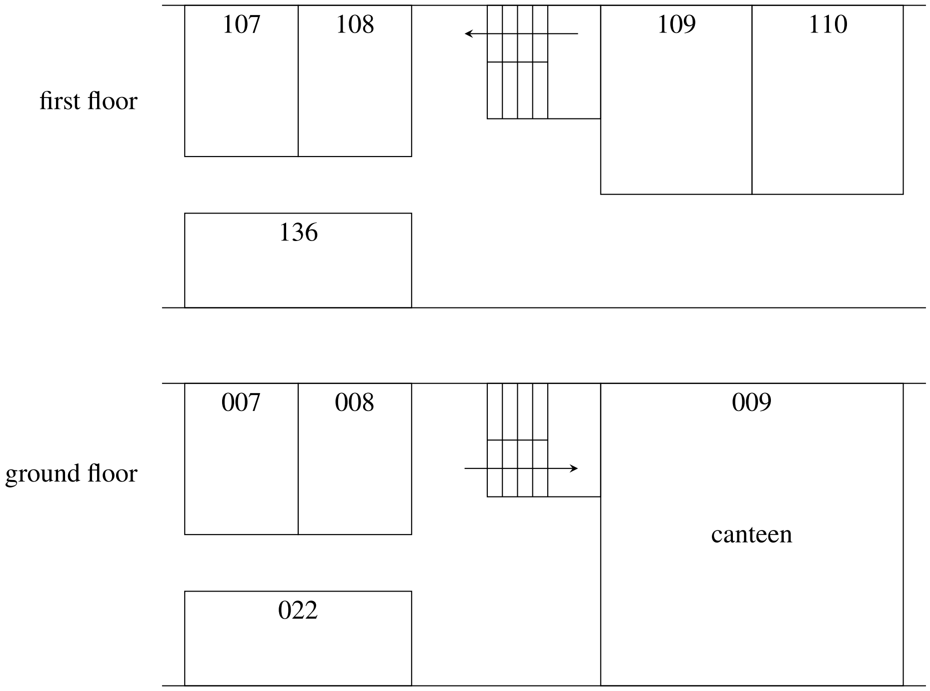


Figure 1: A fragment of a university building floor plan.

many subtypes: IfcBuildingStorey, IfcWall, IfcDoor, IfcOpeningElement, etc. Entities can be arranged in many different, complex ways, thus parsing an IFC file is a nontrivial task.

Systems which analyze BIM models in order to determine where human-accessible spaces are and how they are connected are described in, e.g., (Isikdag et al., 2013), (Tang et al., 2015) or (Xu et al., 2017). (Ferreira et al., 2018) presents a campus navigation system which integrates spatial data extracted from the BIM model with knowledge from other sources, which allows it to handle queries referring to university teachers, e.g. “how can I get to the Professor Smith’s room?”.

Location-based services (Sharma et al., 2017) are another area in the field of indoor navigation. Majority of theoretical research in this area focuses on privacy aspects; among practical applications, systems dedicated for museum environments are common.

For example, MusA (Rubino et al., 2013) combines roles of a navigation system and an electronic guide. This necessarily implies that its internal database contains not only information about spatial structure of the museum’s building, but also about exhibits (names, descriptions, etc.). Its navigation module allows users to either select a single target point, or a path consisting of several points. A path can be specified manually, point by point, or chosen from the list of predefined thematic paths designed by museum’s curators.

It is worth noticing that even when the internal database of the navigation system includes additional semantic data, the user usually cannot fully take advantage of it. For example, MusA knows descriptions associated with exhibits and can display them for the user, but cannot search these descriptions for particular keywords when the user is planning a path.

This paper attempts to propose a solution which gives the user a way of specifying navigation targets on basis of their semantic properties.

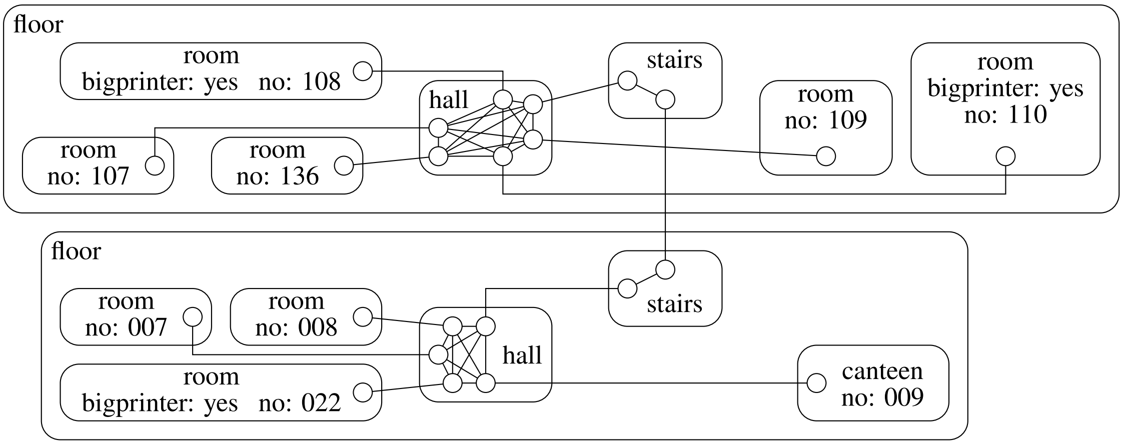


Figure 2: A view showing the spatial hierarchy (edge costs omitted).

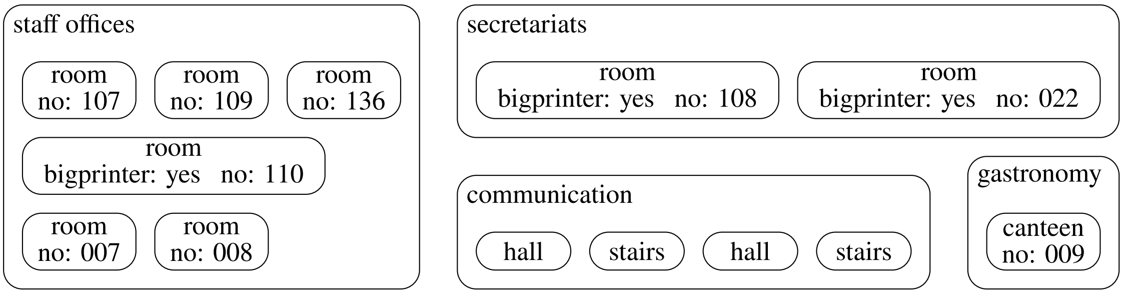


Figure 3: A view showing the functional hierarchy.

3. The Graph Model

Multi-hierarchical graphs proposed in (Palacz et al., 2016) are a result of research on design models presented in e.g. (Grabska et al., 2007, 2013; Strug et al., 2014). Ordinary flat graphs are often inadequate as models of structures encountered in real-world architectural design and engineering applications. Such complex structures are often hierarchical; therefore hierarchical graphs are proposed. They allow for separating components of modelled item from each other and for “hiding” some details in the lower levels of the hierarchy, while retaining the ability to specify relations between parts of different components.

The idea of multiple hierarchies was initially introduced in order to provide an additional spatial hierarchy, which was used to group stair segments and stair landings into staircases. It was necessary because staircases do not fit into the standard spatial hierarchy, which divides a building into separate floors, which are further divided into spaces (rooms, corridors, etc.).

In (Palacz et al., 2016) extra hierarchies were used to represent non-spatial relations between parts of the building. These relations could, in theory, be represented by appropriately labelled graph edges, but using graph hierarchies is more convenient when dealing with relations which by their inherent nature also are hierarchical.

Let us consider a small part of a university building as an example. Fig. 1 displays its floor plan, which includes several rooms belonging to two departments, a student canteen and a staircase. A multi-hierarchical graph which models this building has four hierarchies: spatial, vertical spatial, functional and administrative. A graph with many hierarchies is difficult to arrange on a diagram in a clear way, therefore views of different hierarchies are displayed in separate figures.

The standard spatial hierarchy (Fig. 2) groups together nodes representing spaces belonging

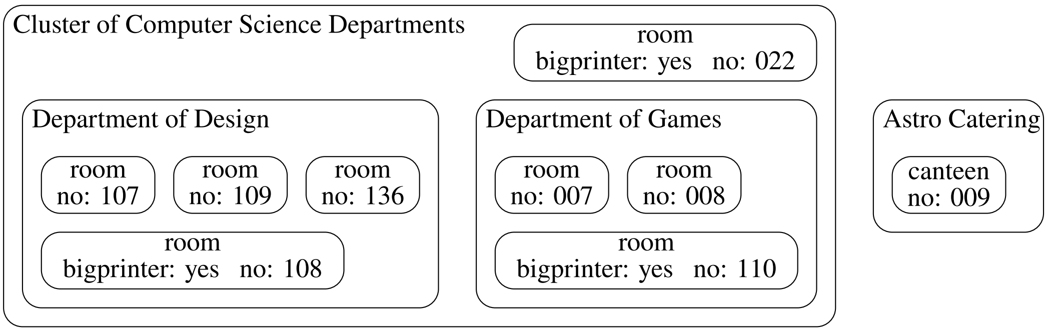


Figure 4: A view showing the administrative hierarchy.

to the same floor and places them inside a node representing that floor. It also contains nodes representing entry/exit points which are used to move between adjacent spaces. These nodes are connected by edges denoting accessibility. The cost of traversing a given edge (that is, time required to walk from one point to the other) is stored in that edge's attribute.

The functional hierarchy (Fig. 3) divides the building into functional areas. All corridors and stairs are assigned as children of the "communication" node, personal rooms of university employees as children of the "staff offices" node, canteens and other places where food can be bought are assigned to the "gastronomy" node, and so on.

The administrative hierarchy (Fig. 4) provides nodes corresponding to the administrative structure of the university (chairs, departments, etc.).

Nodes representing rooms take part in all hierarchies and are presented on all three views. For other nodes it varies, depending on what exactly do they represent. Halls, as common building areas which belong to no department, take part only in spatial and functional hierarchies. Parent nodes which group rooms usually correspond to immaterial entities which make sense only in context of a specific hierarchy, and thus take part only in that hierarchy.

In addition to hierarchies discussed above, the example graph model has node labels and attributes. They are used to store semantic data about building and its spaces, too. As can be seen in Figs. 2–4, information about types of spaces (room, hall, etc.), names given to grouping entities, room numbers and presence of big-format printers is represented.

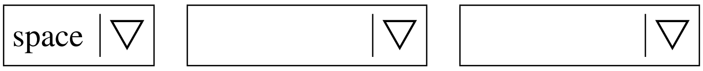
4. Evolution of the Knowledge Base and Target Selection GUI

The navigation system proposed in (Palacz et al., 2016) requires both geometric data and semantic data. A single graph model which stores both is a considerable advantage. It provides the maximum possible degree of integration and eliminates the possibility of inconsistency between two separate databases. A graph is a well-known data structure, therefore developers charged with creating, updating and verifying building models should find that tasks straightforward.

Graphs are also easily extensible. Introducing a new attribute, assigning it to selected nodes and defining attribute values for these nodes does not require any changes in the formal definition of a multi-hierarchical graph. New node hierarchies can be easily introduced, too.

It seems probable that after deploying the navigation system, its users (i.e. students, lecturers and other staff) will propose new kinds of semantic data which should be included in the knowledge base. For example, they may wish to know canteen hours, or the number of seats in a lecture room (and if they have power sockets for laptops), or which equipment is installed in a laboratory.

To understand how this evolution of database contents may proceed we can look at Open-



space ▽ | ▽ | ▽

Figure 5: The initial state of the search form.

StreetMap (OSM), a well-known collaborative mapping project. OSM uses a topological data structure with three basic element types: points, polylines/polygons, and lists of elements. Each element usually has one or more tags, which are pairs consisting of a key and a value. For example, a hypothetical polygon could be be tagged with the “building=house”, “addr:street=Apple Street” and “addr:housenumber=17” tags, which provide the type of object (a single-family house) and its semantic data. So, it should be clear that OSM tags are very similar to labels and attributes used in the multi-hierarchical graph model.

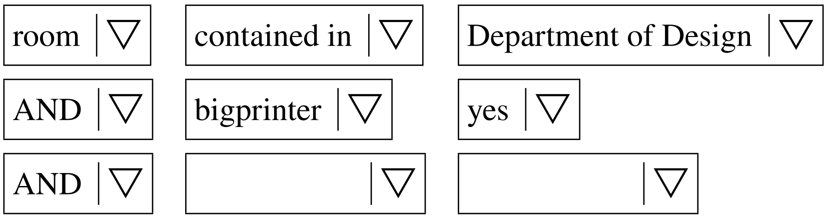
OSM is edited by volunteers, and each of them can invent their own tags. Nonetheless, the OSM community produced a common set of conventions describing which tags should be used for which purposes (OSM user community, 2019). This set of commonly used tags evolved over time. (Mocnik et al., 2017) describes this process and notes that it included introduction of new keys, but also refinement of existing tags. Consider the tag “building=residential”, which is rather general—by introducing additional values “apartments”, “detached”, “terrace”, etc. the map can be made more detailed. Currently there are several dozen frequently used keys, with no more than several dozen possible values per key (this does not include keys with free-form values, like “addr:street”). Let us assume that these upper limits will also hold for the graph model.

As was stated in the introduction, this paper aims to propose a GUI for specifying a navigation target by leveraging semantic data stored in the graph. This GUI needs to be able to adapt to unexpected changes in the knowledge base (introduction of new attributes or hierarchies, changes in attributes’ value ranges).

The navigation system must analyse the graph model. It starts by using spatial hierarchy to find nodes corresponding to basic spaces and makes a list of their labels (which denote their types). Then it checks other hierarchies and makes another list, with labels of group nodes which contain found spaces. It also makes lists with attribute names and their values. Since text strings gathered on these lists will be displayed in the GUI, they must be human-readable and their meaning should be clear to an average user.

Look and feel of the GUI is similar to advanced search forms provided by electronic libraries. We start with an empty form (see Fig. 5). It provides three drop-down list widgets which allow us to specify which type of space we are interested in, and which simple condition this space must fulfill. Leaving all three of them in their initial state is technically allowed, but means that we want to visit any/all spaces in the building.

Widgets must be filled left-to-right. The middle drop-down list is used to specify is we want to check a specific attribute (after selecting its name the right list will offer a choice between values of this attribute) or a hierarchical relation (in that case the right list offers labels of group



room ▽ | contained in ▽ | Department of Design ▽
AND ▽ | bigprinter ▽ | yes ▽
AND ▽ | ▽ | ▽

Figure 6: An example target description.

nodes). When all widgets are filled the form automatically extends, allowing us to specify another simple condition and combine it with the previous one. Fig. 6 shows an example which combines two conditions.

Constructed target description resolves to a set of spaces; it is worth noticing that this set may contain zero, one or more elements. In the case of multiple results, the navigation system will direct its user to the closest target, then ask if he wishes to continue onward to the next one, and so on until all targets are visited or the user commands it to stop.

Target descriptions can be chained together so that the system will guide the user to targets from the first set, then to targets from the second set (unless they were already visited in phase one), and so on.

5. Conclusion

This paper considered an indoor navigation system described in (Palacz et al., 2016), which uses a graph as its main data structure. The graph contains data required for path planning, as well as additional data which semantically describes rooms and other elements of a building. The semantic knowledge could be used to execute complex navigation tasks, but only by implementing a procedure which extracted and processed data appropriate to a given task.

To make the system more useful for ordinary users, this paper proposed a specialized graphical user interface. This GUI adjusts itself dynamically by taking into account types of semantic knowledge which are stored in the graph, and provides a point-and-click form which allows for describing navigation targets on the basis of their semantic properties. Form's complexity is about equal to the "advanced search" forms provided by online libraries and document repositories.

References

Aruba Networks (2019). The Meridian Platform. `https://meridianapps.com/`, accessed on June 1, 2019.

buildingSMART International (2019). IFC introduction. `https://www.buildingsmart.org/about/what-is-openbim/ifc-introduction/`, accessed on June 1, 2019.

Correa, A., Barcelo, M., Morell, A., & Vicario, J. L. (2017). Indoor positioning systems for mass market applications. *Sensors*, 17(8).

Ferreira, J. C., Resende, R., & Martinho, S. (2018). Beacons and BIM models for indoor guidance and location. *Sensors*, 18(12).

Gilliéron, P.-Y., Chazal, V., Flamm, M., von der Mühll, D., & Ruzicka-Rossier, M. (2013). Pedestrian navigation for the benefit of mobility. In A. Nait-Sidi-Moh, M. Bakhouya, J. Gaber, & M. Wack (Eds.), *Geopositioning and Mobility* chapter 7, (pp. 155–201). John Wiley & Sons.

Grabska, E., Łachwa, A., Ślusarczyk, G., Grzesiak-Kopeć, K., & Lembas, J. (2007). Hierarchical layout hypergraph operations and diagrammatic reasoning. *Machine GRAPHICS & VISION*, 16(1/2), 23–38.

Grabska, E., Ślusarczyk, G., & Gajek, Sz. (2013). Knowledge representation for human-computer interaction in a system supporting conceptual design. *Fundamenta Informaticae*, 124(1–2), 91–110.

Isikdag, U., Zlatanova, S., & Underwood, J. (2013). A BIM-oriented model for supporting indoor navigation requirements. *Computers, Environment and Urban Systems*, 41, 112–123.

Köbben, B., van Bunningen, A. H., & Muthukrishnan, K. (2006). Wireless campus LBS: Building campus-wide location based services based on WiFi technology. In E. Stefanakis, M. P. Peterson, C. Armenakis, & V. Delis (Eds.), *Geographic Hypermedia: Concepts and Systems* (pp. 399–408). Springer.

Mocnik, F.-B., Zipf, A., & Raifer, M. (2017). The OpenStreetMap folksonomy and its evolution. *Geo-spatial Information Science*, 20(3), 219–230.

OSM user community (2019). Map features. `https://wiki.openstreetmap.org/wiki/Map_Features`, accessed on March 25, 2019.

Pahlavan, K., Li, X., & Mäkelä, J.-P. (2002). Indoor geolocation science and technology. *IEEE Communications Magazine*, 40(2), 112–118.

Palacz, W., Ślusarczyk, G., Łachwa, A., Strug, B., Paszyńska, A., & Grabska, E. (2016). Towards knowledge-based building navigation with the use of a hierarchical graph model. In *Proceedings of the 23rd EG-ICE International Workshop* (pp. 301–311). Kraków, Poland.

Rubino, I., Xhembulla, J., Martina, A., Bottino, A., & Malnati, G. (2013). MusA: Using indoor positioning and navigation to enhance cultural experiences in a museum. *Sensors*, 13(12).

Sharma, L., Javali, A., Nyamangoudar, R., Priya, R., Mishra, P., & Routray, S. K. (2017). An update on location based services: Current state and future prospects. In *Proceedings of the 2017 International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 220–224).

Strug, B., Grabska, E., & Ślusarczyk, G. (2014). Supporting the design process with hypergraph genetic operators. *Advanced Engineering Informatics*, 28(1), 11–27.

Tang, S. J., Zhu, Q., Wang, W. W., & Zhang, Y. T. (2015). Automatic topology derivation from IFC building model for in-door intelligent navigation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-4/W5.

Xu, M., Wei, S., Zlatanova, S., & Zhang, R. (2017). BIM-based indoor path planning considering obstacles. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W4, 417–423.