Appendix MULTIDIMENSIONAL RELATIONSHIPS RULES

In this section, we define a set of rules (54 rules) that allow us to determine a relationship between two ideas. We first define the rules for the Pattern WHAT and then the rules for Pattern HOW. After each rule, we give two ideas as example of defining the typed relation. The output of the identification of ideas' elements step for each pattern is either a triple, single term (possibly expressed by multiple words) or a sequence of words (multiple words which cannot be reduced to a single term). The rules are written in natural language in order to be comprehensible by reviewers.

We should reminder the reviewers that the matching system uses terminology, structure, linguistic and content-based matchers. These matchers are executed sequentially and the a link to identify between terms depends on each matcher's threshold. Based on experiment of ranging a threshold from 0.55 to 0.95, we fixed a threshold 0.75 for the linguistic-based matcher; a threshold 0.80 for the content-based matcher and a threshold 0.85 for Terminology-based matcher. Finally, the structure-based matcher do not depend on threshold value because the relation between two terms either is defined in Wikidata or not.

Appendix.1 Rules for pattern WHAT:

Before describing the identification rules that identify relationships between ideas. We define:

- If the part (what) of idea1 is composed of <subject S1, predicate R1, object O1> and the part (what) of idea2 is composed only of <SingleTerm W1> Then
  - if we identify that $((S1\ or\ O1)sameAs(W1) \vee (S1\ or\ O1)superclass(W1) \vee (S1\ or\ O1)SimilarTo(W1) \vee (S1\ or\ O1)subclass(W1))$ then
    - it's generalizes when the part (how) of idea1 is the same part of (how) of idea2
      - Window composed of heat-sensitive material (indicates the exit path)
      - window (shows the exit path)
    - it's alternative solution when the part (how) of idea1 is different then the part (how) of idea2
      - Window composed of heat-sensitive material (shows person position)
      - Facade (paints the exit path)
  - else if no such relation detected $(S1orO1)sameAs(W1) \vee (S1orO1)superclass(W1) \vee (S1orO1)SimilarTo(W1) \vee (S1orO1)subclass(W1)$
    - it's similar when the part (how) of idea1 is the same part of (how) of idea2
      - Window composed of heat-sensitive material (indicates the escape route)
      - Smoke (shows the exit path)
    - it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
      - Window composed of heat-sensitive material (sends information to fire department)
      - Smoke (shows the exit path)

  //
- If the part (what) of idea1 is composed of <subject S1, predicate R1, object O1> and the part (what) of idea2 is composed of sequence of words WS1 Then
  - Compute similarity between WS1 and <S1 + O1> if similarity >= threshold
    - it's similar when the part (how) of idea1 is the same part of (how) of idea2
      - The building consists of a variable structure (indicates the escape route)
      - Changeable building structure (shows the exit path)
    - it's alternative solution when the part (how) of idea1 is different then the part (how) of idea2
      - The building consists of a variable structure (indicates the escape route)
      - Changeable building structure (transforms to flying jet)
  - else similarity <threshold
    - it's alternative solution when the part (how) of idea1 is the same part of (how) of idea2
      - The building consists of a variable structure (indicates the escape route)
      - Flying Drones (shows the exit path)
    - it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
      - The building consists of a variable structure (indicates the escape route)
      - Non-flammable suit (that's activates and cover person body)

  //

  We denote SRS, SR, SD, VR, VD, OR, OD, ORS as follows:
- If the part (what) of idea1 of <subject S1, predicate R1, object O1> and the part (what) of idea2 is composed of <S2, R2, O2>
  If $((s_1)similarTo(s_2) \vee (s_1)subclass(s_2) \vee (s_1)superclass(s_2))$ Then == SR
  If $(s_1)synonymTo \vee$ sameAs(s_2) Then SRS Else == SD
  If $(r_1)\ similarTo\ (r_2) \vee (r_1)\ synonymTo\ (r_2)$ Then == VR Else == VD (if antonym is detected then it's not considered)
  If $((o_1)similarTo(o_2) \vee (o_1)\ subclass(o_2) \vee (o_1)superclass\ (o_2))$ Then == OR
  If $(o_1)sameAs(o_2)$ Then ORS Else == OD

(1) SRS VR ORS: means if we have same subjects (SRS), same verbs (VR), same objects (ORS)

, ,

(a) it's duplicate when the part (how) of idea1 is the same part of (how) of idea2
- Window composed of heat-sensitive material (indicates the exit path)
- Window made of heat-sensitive material (shows the exit path)

(b) it's alternative solution when the part (how) of idea1 is different then the part (how) of idea2
- Window composed of heat-sensitive material (indicates the exit path)
- Window made of heat-sensitive material (sends position info)

(2) (RS or SRS) VR OR: means if we have same (SRS) or related subjects (RS), same verbs (VR), related objects (ORS)

(a) it's generalize/specialize when the part (how) of idea1 is the same part of (how) of idea2
- Window composed of heat-sensitive material (indicates the exit path)
- Architecture element made of TCO (shows the exit path)

(b) it's alternative solution when the part (how) of idea1 is different then the part (how) of idea2
- Window composed of heat-sensitive material (indicates the position)
- Architecture element made of TCO (prints exit path on wall)

(3) (RS or SRS) VR OD: means if we have same (SRS) or related subjects (RS), same verbs (VR), different objects (OD)

(a) it's alternative solution when the part (how) of idea1 is the same part of (how) of idea2
- Window composed of heat-sensitive material (indicates the exit path)
- Architecture element made of plants (shows the exit path)

(b) it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
- Window composed of heat-sensitive material (indicates the exit path)
- Architecture element made of plants (sends position information through the smell)

(4) (RS or SRS) VD (OR or ORS)

(a) it's alternative solution when the part (how) of idea1 is the same part of (how) of idea2
- The building consists of a variable structure (indicates the exit path)
- Architecture element cover-up of TCO (shows the exit path)

(b) it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
- The building consists of a variable structure (transforms in to flying bed)
- Architecture element cover-up of TCO (shows the exit path)

(5) (RS or SRS) VD OD

(a) it's disjoint solution when the part (how) of idea1 is the same or different from the part (how) of idea2
- The building consists of a variable structure (covers person body)
- Architecture element cover-up with inflammable material (packs human)
- The building consists of a variable structure (turns into a tunnel)
- Architecture element cover-up with inflammable material (packs human)

(6) SD VR (OR or ORS)

(a) it's alternative solution (or similar) when the part (how) of idea1 is the same part of (how) of idea2
- The room consists of a variable structure (capsules human)
- Refrigerator made of TCO (packs human)

(b) it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
- The room consists of a variable structure (turns into a tunnel)
- Refrigerator made of TCO (packs human)

(7) SD VR OD

(a) it's disjoint when the part (how) of idea1 is the different part of (how) of idea2
- The room consists of a variable structure (covers person)
- Refrigerator made of inflammable material (packs human)

(b) it's alternative solution when the part (how) of idea1 is the same part of (how) of idea2
- The room consists of a variable structure (turns into a tunnel)
- Refrigerator made of inflammable material (packs human)

(8) SD VD OD

(a) it's alternative solution when the part (how) of idea1 is the same part of (how) of idea2
- The room consists of a variable structure (packs human)
- Refrigerator cover-up of inflammable material (packs human)

(b) it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
- The room consists of a variable structure (turns into a tunnel)
- Refrigerator cover-up of inflammable material (packs human)

(9) SD VD (OR or ORS)

(a) it's alternative solution when the part (how) of idea1 is the same part of (how) of idea2
- The room consists of inflammable material (packs human)

  - Refrigerator cover-up of inflammable material (packs human)
 (b) it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
  - The room consists of a inflammable material (absorbs fire)
  - Refrigerator cover-up of inflammable material (packs human)

- If the part (what) of idea1 is composed of <SingleTerm W1> the part (what) of idea2 is composed only of <SingleTerm W2> Then
  - if we identify that $((W1)superclass(W2) \vee (W1)subclass(W2))$ then
    - it's generalize when the part (how) of idea1 is the same part of (how) of idea2
      - Window (indicates the escape root)
      - Structure element (shows the exit path)
    - it's alternative solution when the part (how) of idea1 is different then the part (how) of idea2
      - Window (indicates the escape root)
      - Structure element (transforms to a cloud)
  - if we identify that $((W1)sameTo(W2))$ then
    - it's duplicate when the part (how) of idea1 is the same part of (how) of idea2
      - Window (indicates the escape root)
      - Window (shows the exit path)
    - it's alternative solution when the part (how) of idea1 is different then the part (how) of idea2
      - Window (generates lights)
      - Window (sends information about the position)
  - if we identify that $((W1)similarTo(W2))$ then
    - it's alternative solution when the part (how) of idea1 is the same part of (how) of idea2
      - Window (indicates the exit path)
      - Facade (shows the exit path)
    - it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
      - Window (shows the exit path)
      - Facade (sends information about the position)
  - if we identify that $((W1)disjoint(W2) \vee (W1)notsimilar{<}threshold(W2))$ then
    - it's alternative solution when the part (how) of idea1 is the same part of (how) of idea2
      - Window (indicates the exit path)
      - Smoke (shows the exit path)
    - it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
      - Smoke (shows the exit path)
      - Window (sends information about the position)

- If the part (what) of idea1 is composed of <SingleTerm W1> the part (what) of idea2 is composed of <sequence of word WS2> Then
  - Compute similarity between W1 and WS2 if similarity =>threshold Then
    - it's similar when the part (how) of idea1 is the same part of (how) of idea2
      - Intelligent rescue system (shows the exit path)
      - Mobile app (indicates the escape root)
    - it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
      - Intelligent rescue system (paints exit path on wall)
      - Mobile app (indicates the escape root)
  - else similarity <threshold
    - it's alternative solution when the part (how) of idea1 is the same part of (how) of idea2
      - Intelligent rescue system (shows the exit path)
      - Window (indicates the escape root)
    - it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
      - Intelligent rescue system (shows the exit path)
      - Window (transforms to cloud)

- If the part (what) of idea1 of <sequence word SW1> and the part (what) of idea2 is composed of <sequence word SW2> Then
  - Compute similarity between WS1 and WS2 if similarity =>threshold Then
    - it's similar when the part (how) of idea1 is the same part of (how) of idea2
      - Flying robots (move person to outside)

- · Mobile rescue-drone (guide person to exit path)
  - ∗ it's disjoint solution when the part (how) of idea1 is different then the part (how) of idea2
    - · Flying robots (move person to outside)
    - · Mobile rescue-drone (docks the wall into escape root)
- – else similarity <threshold
  - ∗ it's alternative solution when the part (how) of idea1 is the same part of (how) of idea2
    - · changeable building structure (docks into escape root)
    - · Mobile rescue-drone (docks the wall into escape root)
  - ∗ it's disjoint when the part (how) of idea1 is different then the part (how) of idea2
    - · changeable building structure (docks into escape root)
    - · Mobile rescue-drone (move persons outside)

Appendix.2 Rules for pattern HOW:

- If the part (how) of idea1 of <subject S1, predicate R1, object O1> and the part (how) of idea2 is composed of <subject S2, predicate R2, object O2> Then
  If $((s_1)sameAs(s_2) \vee (s_1)similarTo(s_2) \vee (s_1)subclass(s_2) \vee (s_1)superclass(s_2))$ Then == SR Else == SD
  If (r_1) $synonymToorsimilarTo$ (r_2) Then == VR Else == VD
  If $((o_1)sameAs(o_2) \vee (o_1)similarTo(o_2) \vee (o_1)$ $subclass(o_2) \vee (o_1)superclass$ $(o_2))$ Then == OR Else == OD
  (1) RS VR OR then it same how
  - Window indicates people
  - Facade shows person

  (2) RS VR OD then it different how
  - Window indicates path of exit
  - Facade shows person position

  (3) RS VD OR then it same how
  - Window indicates path of exit
  - Facade paints path of exit

  (4) RS VD OD then it different how
  - Window indicates person position
  - Facade paints path of exit

  (5) SD VR OR then it same how
  - Window indicates path of exit
  - smoke shows path of exit

  (6) SD VR OD then it different how
  - Window indicates position
  - smoke shows path of exit

  (7) SD VD OD then it different how
  - net works as slide
  - Smoke shows the path of exit

  (8) SD VD OR then it different how
  - structure transforms to fire escape
  - Smoke shows the path of exit
- If the part (how) of idea1 is composed of <subject S1, predicate R1, object O1> and the part (how) of idea2 is composed only of SingleTerm <W1> Then if compute similarity between S1 or O1 and W1 >= threshold The same how else different how
  - signal is On outside
  - information
- If the part (how) of idea1 is composed of <subject S1, predicate R1, object O1> and the part (how) of idea2 is composed of sequence of words <SW1> Then if compute similarity between <subject S1, predicate R1, object O1> and SW1 >= threshold The same how else different how
  - signal is On outside
  - Save navigation to the exit
- If the part (how) of idea1 is composed of <only word W1> and the part (how) of idea2 is composed only SingleTerm <W2> Then if W1 subclass of W2 or superclass or similar or same then same how else different how
  - exit path
  - escape route

- If the part (how) of idea1 is composed of <only word W1> and the part (how) of idea2 is composed of sequence of words <SW2> Then if compute similarity between <W1> and SW2 >= threshold The same how else different how
  - position
  - Save navigation to the exit
- If the part (how) of idea1 of <sequence of words SW1> and the part (how) of idea2 is composed of sequence of words <SW2> Then if compute similarity between <SW1> and SW2 >= threshold The same how else different how
  - sending and receiving info
  - Save navigation to the exit