Textual Entailment Graph Dataset - ENGLISH

Excitement Project

V2.0 - 31 December 2014

Introduction

We present the first gold standard dataset of Textual Entailment Graphs (TEG), which was constructed in order to introduce the task of automatic TEG generation to the community. A TEG is a directed graph where each node is a complete natural language text (textual fragment) f_i and each edge (f_i, f_j) represents an entailment relation from f_i to f_j. A textual entailment (f_i, f_j) holds if the meaning of f_i implies the meaning of f_j, according to the standard definition of textual entailment which states that f_i entails f_j if, typically, a human reading f_i would infer that f_j is most likely true.

Given a set of textual fragments (graph nodes), the task of constructing a TEG is to recognize all the entailments among the fragments, i.e. deciding which pairs of nodes are connected by directional edges. The main difference between this task and the traditional Recognizing Textual Entailment (RTE) task is that the text pairs are not independent. The nodes in the graph are inter-connected via entailment edges, which should not represent contradicting decisions. For example, if the edges (*u,v*) and (*v,w*) are in the graph, then the edge (*u,w*) is implied by transitivity. In the TEG, bi-directional entailments (i.e. equivalence between nodes) are encoded as two separate edges (*x,y*) and (*y,x*).

Our motivating scenario was text exploration - in particular the analysis of customer dissatisfaction - and the dataset was constructed for a text collection of email feedbacks sent by customers of a railway company. Textual fragments were manually extracted in such a way that each fragment contains a single proposition where a customer states a reason for dissatisfaction with the company, like *"No vegetarian snacks in the dining car"* or *"There's not enough food selection on train"*.

To reduce the annotation complexity, as well as to allow evaluation of TEG generation for particular subtopics (clusters) of the target collection, fragments were manually clustered into 29 subtopics such as "legroom", "internet connection", "food choice". Then, given the textual fragments within each cluster as input, a gold standard TEG was built for each of the clusters as a pipeline of two separate sub-tasks, namely *(i)* further decomposing each input fragment and constructing its individual textual entailment graph - which we term fragment graph - and *(ii)* merging the fragment graphs into a single integrated TEG.

1) Construction of Fragment Graphs. Textual fragments can be further simplified to provide higher generalization ability, thus increasing the probability of recognizing entailing texts in the collection, and provide a richer hierarchical structure. For example, the fragment *"No vegetarian snacks in the dining car"* can be generalized in its subfragment *"no vegetarian snacks"*. Such generalization can be effectively performed automatically, if grammatical modifiers, i.e. tokens that can be removed from a fragment without affecting its comprehension, as well as the dependencies between them are specified. For each textual fragment in the clusters, modifiers were manually annotated and then an automatic procedure generated its corresponding subfragments by incrementally removing modifiers until no modifiers were left. In addition, entailment relations were automatically induced following the principle that a more specific text (i.e. containing more modifiers) entails a more generic one (i.e. containing less modifiers). As a result, an entailment graph of the corresponding fragment was constructed.

2) Merging Fragment Graphs into integrated TEGs. The individual fragment graphs within each cluster were manually merged into a single TEG by means of identifying entailment relations between them. During the annotation process transitivity control was performed in order to detect violations and resolve them. As a result, the dataset contains 29 consistent TEGs.

Given the complex scenario of TEG construction, we offer two datasets corresponding to the two main TEG creation tasks:

1) annotation of modifiers in textual fragments

2) fragment graph merging.

Data Format Documentation

1) MODIFIER_ANNOTATION_DATASET_ENG_V2.0

This directory contains 29 directories, one for each identified cluster.

Each cluster directory contains 2 directories:

- ***Interactions***, containing the original customer interactions from which the fragments were extracted;

- ***annotatedFragments***, containing an xml file where the modifier annotation of all the fragments is represented with the following format.

```
<Fragments fragmentNum="19">
   <fragment id="418508.txt_5_0" modality="TRUE" explicit_context="418508.txt"
             implicit_context="" focus="expresso" implicit_focus="">
        <original_text>
              <text>
                There should be a proper expresso machine , not that horrible nespresso
              </text>
              <token id="0">There</token>
              <token id="1">should</token>
              <token id="2">be</token>
              <token id="3">a</token>
              <token id="4">proper</token>
              <token id="5">expresso</token>
              <token id="6">machine</token>
              <token id="7">,</token>
              <token id="8">not</token>
              <token id="9">that</token>
              <token id="10">horrible</token>
              <token id="11">nespresso</token>
        </original_text>
        <correct_text>
           There should be a proper espresso machine , not that horrible nespresso
        </correct_text>
        <modifiers_list>
             <modifier id="0">
                 <token_anchor id="8"/>
                 <token_anchor id="9"/>
                 <token_anchor id="10"/>
                 <token_anchor id="11"/>
             </modifier>
        </modifiers_list>
   </fragment>
   <fragment...
   </fragment>
   ...
</Fragments>
```

Note that:
- `fragmentNum=""`: the total number of fragments in the cluster (and in the xml)
- for each `<fragment>` element:
    - `id=""`: the ID of the fragment
    - `modality="TRUE/FALSE"`: if the fragment expresses modality
    - `explicit_context=""`: the original interaction from which the fragment has been taken
    - `implicit_context=""`: in case the fragment assumes implicit information, which is relevant for its correct interpretation but is not explicitly expressed in the interaction from which it has been taken, such implicit information is made explicit through a free list of keywords
    - `focus=""`: the entity or fact/event the customer is complaining about in the fragment
    - `implicit_focus=""`: the focus of the fragment when it is not expressed by any word in the fragment
    - `<original_text>`: the fragment as it appears in the original interaction, represented both as plain text `<text>` and tokenized `<token id=""> </token>`
    - `<correct_text>`: contains the fragment corrected by the annotator when the original one is not a grammatically complete and well-formed sentence
    - `<modifiers_list>`: contains the annotation of modifiers
        - `<modifier id="">` : identifies the modifier
        - `<token_anchor id=""/>` : the list of the tokens in the fragment composing the modifier

2) ENTAILMENT_GRAPH_DATASET_ENG_V2.0

This directory contains 29 directories, one for each identified cluster.

Each cluster directory contains 3 directories:

- ***Interactions***, containing the original customer interactions from which the fragments were extracted;

- ***FragmentGraphs***, containing the fragment graphs (FG) created for each single fragment. FGs are created automatically on the basis of modifiers' annotation, as explained in the Introduction.
  Each FG is represented as an XML file with the following structure:

```
<F_entailment_graph nodeNum="2">
    <node id="418508.txt_5_0" modality="TRUE" explicit_context="418508.txt"
          implicit_context="" focus="expresso" implicit_focus="">
        <original_text>There should be a proper expresso machine , not that horrible
                       nespresso</original_text>
        <correct_text>There should be a proper espresso machine , not that horrible
                      nespresso</correct_text>
    </node>
    <node id="418508.txt_5_1" modality="TRUE" explicit_context="418508.txt"
          implicit_context="" focus="expresso" implicit_focus="">
       <original_text>There should be a proper expresso machine ,</original_text>
       <correct_text>There should be a proper espresso machine</correct_text>
    </node>
    <edge id="418508.txt_5_0-418508.txt_5_1"
          source="418508.txt_5_0" target="418508.txt_5_1"/>
</F_entailment_graph>
```

Note that:
- `nodeNum=""`: the total number of nodes in the FG
- for each `<node>` element:
  - `id=""`: the ID of the node
  - `modality="TRUE/FALSE"`: if the node text expresses modality
  - `explicit_context=""`: the original interaction from which the node text comes from
  - `implicit_context=""`: in case the node text assumes implicit information, which is relevant for its correct interpretation but is not explicitly expressed in the interaction from which it comes from, such implicit information is made explicit through a free list of keywords
  - `focus=""`: the entity or fact/event the customer is complaining about in the node text
  - `implicit_focus=""`: the focus of the node text when it is not expressed by any word in the text
  - `<original_text>`: the node text as it appears in the original interaction
  - `<correct_text>`: the node text corrected by the annotator when the original one is not a grammatically complete and well-formed sentence
- for each `<edge>` element:
  - `id=""`: the ID of the edge
  - `source=""`: the ID of the source node (the entailing text)
  - `target=""`: the ID of the target node (the entailed text)

- ***FinalMergedGraph***, containing the final entailment graph for that cluster obtained from merging the single FGs. The final TEG is represented as an XML file with the following structure. Note that bi-directional entailments - e.g. equivalence between nodes - are encoded as two separate edges (see below the two edges connecting nodes `id="237090.txt_2_0"` and `id="522350.txt_3_0"`).

```
<F_entailment_graph nodeNum="31">
    <node id="237090.txt_2_0">
        <original_text>More choice of drink</original_text>
        <correct_text></correct_text>
    </node>
    <node id="522350.txt_3_0">
        <original_text>better drinks selection</original_text>
        <correct_text></correct_text>
    </node>
    <node id="63301.txt_1_1">
        <original_text>You could bring back champagne</original_text>
        <correct_text></correct_text>
```

```
    </node>
    <node id="305036.txt_1_0">
        <original_text>there is no more champagne on offer</original_text>
        <correct_text></correct_text>
    </node>
    ...
    <edge id="522350.txt_3_0-237090.txt_2_0"
          source="237090.txt_2_0" target="522350.txt_3_0"/>
    <edge id="237090.txt_2_0-522350.txt_3_0"
          source="522350.txt_3_0" target="237090.txt_2_0"/>
    <edge id="500967.txt_6_3-187987.txt_4_0"
          source="187987.txt_4_0" target="500967.txt_6_3"/>
    <edge id="228464.txt_2_1-228464.txt_2_0"
          source="228464.txt_2_0" target="228464.txt_2_1"/>
    ...
</F_entailment_graph>
```

Note that:

- `nodeNum=""`: the total number of nodes in the final entailment graph
- for each `<node>` element:
    - `id=""`: the ID of the node
    - `<original_text>`: the node text as it appears in the original interaction
    - `<correct_text>`: the node text corrected by the annotator when the original one is not a grammatically complete and well-formed sentence
- for each `<edge>` element:
    - `id=""`: the ID of the edge
    - `source=""`: the ID of the source node (the entailing text)
    - `target=""`: the ID of the target node (the entailed text)

Contact Information

For further information about this data release, contact the following people:

Luisa Bentivogli (FBK) bentivo@fbk.eu
Bernardo Magnini (FBK) magnini@fbk.eu
Lili Kotlerman (BIU) lili.dav@gmail.com