UNIVERSITY OF SÃO PAULO

School of Arts, Sciences and Humanities

Technical Report PPgSI-003/2013

ResDial – Coding Description (v.1.0)

Norton Trevisan Roman

April - 2013

Technical Report Series

PPgSI-EACH-USP. Arlindo Béttio St. 1000 - Ermelino Matarazzo - 03828-000.
São Paulo, SP. Brazil.
TEL: 55 (11) 3091-8197 http://www.each.usp.br/ppgsi

ResDial – Coding Description (v.1.0)

Norton Trevisan Roman[1]

[1]School of Arts, Sciences and Humanities
University of São Paulo (EACH-USP)
Av. Arlindo Béttio, 1000 – 03828-000 – Ermelino Matarazzo, São Paulo – SP – Brazil

norton@usp.br

Abstract. *This document describes the XML tags used in the codification of ResDial – a set of human produced dialogue summary corpora. It corresponds to the English version of the PPgSI-001/2012 report.*

1. Introduction

Corresponding to a set of corpora of human produced dialogue summaries, Resdial repository currently contains two corpora. The first of them was produced from automatically generated dialogues [Roman et al. 2006a], while the second came from movie script dialogues [Roman et al. 2006b]. Even though the summarised dialogues are artificial in nature, both corpora differ in that, in the first set, dialogues were generated by a computational system, whereas in the second they were written by humans.

Whatever the set, source dialogues illustrate buying-selling interactions (*i.e.* interactions between a seller and a prospective buyer), with the automatically generated dialogues limited to car selling, while the others deal with selling goods in general. Both dialogue sets also share the characteristic of portraying situations in which either one, both, or none of the dialogue participants is impolite.

For the summaries, four dialogues were chosen from each source, so that the first dialogue illustrates a situation where the client is rude, while the second presents a situation where the vendor is rude, being the remaining dialogues neutral (*i.e.* portraying situations where no party is rude). Each dialogue set was then presented to groups of volunteers, who should summarise them under one out of three viewpoints (vendor, client and a neutral observer).

In order to reduce the risk of biasing the data, summarisers were randomly assigned to one of the three aforementioned categories. Furthermore, each set was split into two other subsets: one whose summarisers would have no constraint in summary size, and another where summarisers had to produce summaries no longer than 10% of the number of words in the source dialogue. Each volunteer should then summarise all four dialogues, under a given viewpoint, using no more than 10% of the number of words in the source dialogue (whenever appropriate).

As an attempt to standardise the application of coding schemes to the available corpora, a set of XML tags was defined which should be applied not only to the existing corpora but also to future corpora that may be stored in ResDial. In this report[1], this set is described in detail (while a summary may be found in Appendix A), thereby allowing not only for the codification of new corpora, but also for the development of tools for the analysis and annotation of the existing ones.

[1]This report corresponds to the English version of the PPgSI-001/2012 report [Roman 2012].

2. Corpora Organisation

In order to make clear the use of stand off annotation, according to which annotation and source are kept in separate files somehow linked to each other [Ide and Brew 2000], every corpus in ResDial is stored as a UTF-8 encoded plain text, only holding information for classification purposes (see Section 2.2.1 for details). Besides, annotations in ResDial make a clear distinction between the annotation tasks related to corpus structure and those related to the elements within that structure.

As such, ResDial tags can be split up in document identification tags, unit segmentation tags, and classification tags (according to some user defined annotation scheme). The fact that the user can, in a first moment, segment the corpus and, in the sequence, apply some annotation scheme to it presents the advantage of allowing for the use of some tools, from both inside and outside the repository, for statistical calculations, such as inter-annotator agreement, for example, getting specific values both for the definition of the basic unit of annotation and the application of the annotation scheme itself.

This practice, in turn, leads to a reduction in the ambiguity of the obtained results, in that one does not know if some negative result can be owed, for instance, to flaws in the applied annotation scheme, or in the definition of the basic units. Besides, the separation between structuring tasks and annotation tasks makes it possible to develop tools that are more specific to the annotation at hand and, consequently, simpler to use, when compared to more general tools, such as those presented in [Orăsan 2003, Ogren 2006, O'Donnell 2008, Verhagen 2010].

2.1. Directory Tree

Inside ResDial, documents are kept in separate files, with each corpus stored in a directory of its own (defined by the user). Each document is assigned a unique identifier (within its holding corpus), which must also be used in the corresponding file name. The file name must begin with "src", following the pattern "src_[file identifier]_[corpus identifier]_[document author identifier].xml". Likewise, it is also possible to include source texts to the corpus (such as the dialogues used by the summarisers, for example), as if they were a second corpus. In this case, their storage is identical to the main corpus.

The segmentation of a plain document is kept in a separate directory (defined by the user), thereby permitting that different segmentations may be applied to the same source text. Since every segmentation scheme has an identifier, the name of the file keeping the segmented document must also contain the identification of who segmented it (which might represent the opinion of the majority of annotators, for example), following the pattern "seg_[file identifier]_[annotation scheme identifier]_[source corpus identifier]_ [source document identifier in the corpus]_[segmenter identifier].xml" (files resulting from multiple segmenters must have the character 'M' in their identifier, as indicated in Section 2.2.5).

Annotated documents are also kept in separate directories, defined by the user. In this case, files used to store an annotation must be named according to the pattern "ann_[file identifier]_[annotation scheme identifier]_[segmentation scheme identifier]_[annotated document identifier (in the segmentation scheme)]_[annotator identifier].xml". It is worth mentioning that annotations are applied to segmented texts, instead of plain texts.

Finally, given that at any stage in the annotation process one can rely on multiple annotators or participants, each directory admits a "participant" sub-directory, in which all files with information about the participants that produced the directory's content are stored. In this case, each annotator has a separate file, named according to the pattern "part_[participant identifier]_[identifier to the corpus to which he contributed (plain, segmentation, or classification)].xml".

Hence, and assuming that the corpus to be annotated is kept in a directory named "corpus2" (with "corpus1" holding some additional corpus), that each document segmentation is kept in "segmentation1", and that the directory holding the annotations is defined as "scheme1", a possible directory tree would be, for example (notice that, in this example, both segmentation and classification were done by different annotators, as determined by their identifiers):

- corpus1
  - src_0001_c01_null.xml
  - src_0002_c01_null.xml
  - ...
- corpus2
  - src_0001_c02_null.xml
  - src_0002_c02_null.xml
  - ...
- segmentation1
  - seg_0001_S01_c02_0001_M.xml
  - seg_0002_S01_c02_0002_M.xml
  - ...
  - participant
    - part_an001_S01.xml
    - part_an002_S01.xml
- scheme1
  - ann_0001_E01_S01_0001_a01.xml
  - ann_0002_E01_S01_0001_a02.xml
  - ann_0003_E01_S01_0001_a03.xml
  - ann_0004_E01_S01_0002_a01.xml
  - ann_0005_E01_S01_0002_a02.xml
  - ann_0006_E01_S01_0002_a03.xml
  - ...
  - participant
    - part_a01_E01.xml
    - part_a02_E01.xml
    - part_a03_E01.xml

Take, for example, the files in the "segmentation1" directory. In these files, "0001" identifies the document in this folder, "S01" identifies the segmentation scheme used in the process, "c02" represents the plain corpus taken for a source, "0001" identifies the plain document that was segmented, and "M" means that this document reflects the opinion of multiple annotators (see Section 2.2.5). "null" values in the author field (like in the plain text documents above) indicate that the information about the document author is irrelevant or could not be determined.

2.2. Tags Specification

The separation of the possible actions within the corpora in structure annotation tasks and element annotation tasks is accomplished by three sets of tags: identification tags, segmentation tags and classification tags. In addition, a fourth set is used to characterise each corpus' annotators or participants. Whatever the set, comments may be added as in any XML file, with "<!-- This is a comment -->". In what follows, each of these sets will be discussed in more detail.

2.2.1. Identification Tags

In the root of each plain document in ResDial there is a <plainDocument> tag, along with its corresponding </plainDocument> (at the end of the file). At this stage, each document must hold, at least, tags <info type="type" value="value">, in which "type" is an identifier to the type of information being presented (defined by the corpus owner), while "value" is the information proper, along with tags <text> and </text>, which must hold the plain text of the document.

As an example, consider the corpora currently available at ResDial, described in Section 1. In these corpora, relevant information to each document are its identifier and source dialogue, along with the identifier of the corpus that holds the source dialogue, and the identifier to the corpus that contains the document itself, resulting in the file illustrated in Figure 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<plainDocument>
    <info type="id" value="001">
    <info type="corpus" value="c01">
    <info type="source" value="d1">
    <info type="source-corpus" value="f01">
    <text>
        Text of the document...
    </text>
</plainDocument>
```

Figure 1. Codification of a plain document.

It is worth mentioning that the same codification used in the plain document can be used in any other (plain) document. Thus, in this example, both source dialogues and their summaries would be stored in files with the "src" extension, codified according to the example above (even though both are in different directories).

2.2.2. Segmentation Tags

Before any annotation is applied to a plain document, one has to define the text span to which it is to be applied, *i.e.* its basic unit of annotation [van der Vliet et al. 2011, Rodrigues et al. 2012]. These units, in turn, cannot always be defined independently, there usually being some overlapping between them, in that the same portion of text spans over two independent units, along with the embedding of units, *i.e.*, units defined inside other units, and which do not depend on their host unit.

Along with these phenomena, it is also desirable to have basic units allow for [Reidsma et al. 2005, Rodrigues et al. 2012]:

- Interleaving, in that two or more segments, although independent, present interleaved parts; and
- Discontinuities, in that two non contiguous text spans belong to the same unit.

To deal with such requirements, each segmented document in ResDial is kept in a file of its own. This document, in turn, must be encapsulated within <document> and </document> tags. Here too it is possible to have <info> tags, as in the plain document codification (Section 2.2.1), in order to register all necessary information about the segmented document.

Each unit in the segmented document is encapsulated by <unit> and </unit> tags, where each <unit> and corresponding </unit> must necessarily present an identification field, becoming <unit id="identifier"> and </unit id="identifier">. With this field, it is possible to define embedded units, such as those illustrated in Figure 2, so that one can determine to which <unit> some </unit> corresponds. Also, the identification field can be manipulated to allow for discontinuous and interleaved units. To do so, all one has to do is to give two units the same identifier, so they can be interpreted as parts of a single unit.

```
<document>
    <info type="id" value="01">
    <info type="scheme" value="S1">
    <info type="source" value="039">
    <info type="source-corpus" value="c01">
    <info type="annotator" value="M">
    <unit id="0001">This is a text</unit id="0001">
    <unit id="0002">that, <unit ind id="0003">however short,
    </unit ind id="0003"> allows the identification of embeddings
    </unit id="0002">
</document>
```

Figure 2. Splitting the text "This is a text that, however short, allows the identification of embeddings" in basic units.

The embedding of units can happen both between related and unrelated units (and how this is interpreted depends solely on the adopted segmentation scheme). For the former, all one has to do is to write a <unit> tag inside another. For the later, independence is made clear with the addition of the word "ind" to the tag, leading to <unit ind id="identifier"> and </unit ind id="identifier"> tags, as shown in Figure 2.

The need for embedding, overlapping and discontinuities, on the other hand, renders the segmentation scheme non XML compliant, given the lack of a plain representation, which builds into a tree-shaped hierarchy of tags [Krauthammer et al. 2002]. With no other alternative capable of satisfying this project's needs, this "relaxation" of XML definition was found necessary.

2.2.3. Corpus Classification Tags

An annotation (or classification) file must begin with <annotation> and end with </annotation>. As with other files, classification files can also have <info> tags, used to determine the file identifier, its annotator, and the segmented document classified by this file[2], along with any other information that one finds necessary (depending on the adopted annotation scheme).

The classification proper is carried out by <mark unit="annotated unit"> and </mark> tags. Each <mark> tag, in turn, can hold a variable number of <ann type="category" value="value"> tags. Thus, while <mark> links the annotation to its annotated unit (defined in the segmented document pointed out by <info>), <ann> corresponds to the application of the annotation scheme. In this case, each different category in the annotation scheme is assigned an <ann> tag. The category's name is then stored in the tag's "type" field, whereas the value attributed by the annotator is kept in "value".

Figure 3 shows the sample file presented in Figure 2, annotated according to the scheme defined in [Roman and Carvalho 2010]. In this figure, one sees, amongst others, the annotation file identifier and the annotator's identifier, along with the identifier to the file with the source text (split up in basic units of annotation). In the sequence, each unit in the file is annotated (with <mark> tags) according to three dimensions (described in [Roman and Carvalho 2010]), as indicated by <ann> tags.

It is worth mentioning that this codification, just like the one used in the plain document, complies with XML format. Hence, and as a demand of the ResDial repository, the only codification scheme that does not comply with XML is the one used in the text segmentation process.

2.2.4. Participant Characterisation Tags

Since, in general, all stages in the development of a corpus (*i.e.*, collection, segmentation and annotation) are made by more than one person, it might be interesting to the researcher keep some information about each participant, such as gender, age and educational attainment, amongst others. This information is kept in files stored in a subdirectory inside the corpus to which the annotator contributed, with one file per annotator. Each file, in turn, must begin with the <participant> tag, ending with </participant>. Inside the file, only <info> tags are allowed, which store relevant information to the researcher, as illustrated in Figure 4.

2.2.5. Multiple Identifiers

Sometimes, resulting files may come not from a single annotator, but from many (by taking the opinion of the majority, for example). Such cases are characterised by the existence of multiple "<info type="annotator"...>" tags. Although not affecting the annotation itself, the existence of multiple annotators demands a change in the file naming

[2]It is worth mentioning that classifications are assigned to minimum units, *i.e.*, to the results of the segmentation of a plain text.

```
<?xml version="1.0" encoding="UTF-8"?>
<annotation>
    <info type="id" value="a01">
    <info type="scheme" value="A1">
    <info type="annotator" value="an1">
    <info type="source" value="01">
    <info type="source-corpus" value="S1">
    <mark unit="0001">
        <ann type="CATE" value="Neutral report">
        <ann type="INTE" value="">
        <ann type="CONS" value="">
    </mark>
    <mark unit="0002">
        <ann type="CATE" value="Neutral report">
        <ann type="INTE" value="">
        <ann type="CONS" value="">
    </mark>
    <mark unit="0003">
        <ann type="CATE" value="Neutral report">
        <ann type="INTE" value="">
        <ann type="CONS" value="">
    </mark>
</annotation>
```

Figure 3. Classification of the units in Figure 2.

```
<?xml version="1.0" encoding="UTF-8"?>
<participant>
    <info type="id" value="annot01">
    <info type="gender" value="m">
    <info type="age" value="20-30">
</participant>
```

Figure 4. An annotator's characterisation.

pattern, with the use of the character "M" instead of the annotator's identifier. This kind of situation, however, is not restricted to annotators only, being extended to any other tag to which there is a need for multiplicity.

Finally, since files built from many annotators usually represent the opinion of the majority, an extra tag must be added to the file: "<info type="multiple" value="majority">", or "<info type="multiple" value="mode">", for the cases where majority must be absolute (*i.e.* over 50% of the annotators) or just reflect the most commonly assigned classification.

3. Example of Usage

As an example, consider a summary taken from the corpus collected as described in [Roman et al. 2006a], and annotated according to the scheme described in [Roman and Carvalho 2010]. Within ResDial, plain summaries in the corpus are stored in different files. In this example, the summary is stored as illustrated in Figure 5, in a file named "src_039_ c01_sum01 .xml" (the file with the text of the dialogue was removed, for the sake of simplification).

```
<?xml version="1.0" encoding="UTF-8"?>
<plainDocument>
    <info type="id" value="039">
    <info type="corpus" value="c01">
    <info type="source" value="d2">
    <info type="source-corpus" value="f01">
    <info type="viewpoint" value="customer">
    <info type="constraint" value="free">
    <info type="summariser" value="sum01">
    <text>
        Waiting was a bit long, but finally somebody served
        me. I asked him to tell me more about the car. The
        vendor told me about its main characteristics, showed
        me its interior, luggage compartment and other things.
        An apparently nice and elegant car, and safe too, but
        way too expensive, which, along with the little will
        to serve me, made me come back home without buying it.
    </text>
</plainDocument>
```

Figure 5. Plain summary 039 (translated version).

In this file, along with the information usually necessary for corpus maintenance (*i.e.* identifiers to the source dialogue and its corpus, as well as to the file with the summary and its corpus), there are also other three data, which are specific to the annotation scheme and corpus at hand: *viewpoint*, *constraint* and *summariser*. In this case, *viewpoint* corresponds to the point of view under which the summary was written, while *constraint* determines if the summariser was free to produce a free length text, or should bind to the limit of 10% of the number of words of the source dialogue, as mentioned in Section 1. *Summariser*, on the other hand, refers to the person that produced the summary placed between <text> and </text>, whose details are stored in a file inside "participant"– a sub-directory of the directory with the summary in Figure 5.

The segmentation in basic units of annotation is shown in Figure 6, which represents the file "seg_01_S1_c01_039_M.xml". In this figure, the chosen unit is the clause. It should be noted that unit 412 carries an embedded unit (413), thereby resulting in an ill-formed XML. Figures 7 and 8, in turn, show the annotation of the file in Figure 6, which corresponds to the classification of the summary by some annotator (ann01). In this case, the annotation is kept in a file named "ann_01_A1_S1_01_ann01.xml".

Last, the annotator responsible for the results in Figures 7 and 8 is characterised, within the "participant" sub-folder, in a file named "part_ann01_A1.xml", as illustrated in Figure 9. Notice that his identifier – ann01 – matches that of the "annotator" field in Figure 7. Likewise, the summariser responsible for the file in Figure 5 has a similar record in the plain corpus folder (*i.e.* "c01").

```
<document>
    <info type="id" value="01">
    <info type="scheme" value="S1">
    <info type="source" value="039">
    <info type="source-corpus" value="c01">
    <info type="annotator" value="M">
    <unit id="0400">Waiting was a bit long,</unit id="0400">
    <unit id="0401">but finally somebody served me.</unit
    id="0401">
    <unit id="0402">I asked him</unit id="0402">
    <unit id="0403">to tell me more about the car.</unit
    id="0403">
    <unit id="0404">The vendor told me about its main
    characteristics,</unit id="0404">
    <unit id="0405">showed me its interior,</unit id="0405">
    <unit id="0406">luggage compartment</unit id="0406">
    <unit id="0407">and other thing.</unit id="0407">
    <unit id="0408">An apparently nice</unit id="0408">
    <unit id="0409">and elegant car,</unit id="0409">
    <unit id="0410">and safe too,</unit id="0410">
    <unit id="0411">but way too expensive,</unit id="0411">
    <unit id="0412">which, <unit ind id="0413">along with
    the little will to serve me,</unit ind id="0413"> made
    me come back home</unit id="0412">
    <unit id="0414">without buying it.</unit id="0414">
</document>
```

Figure 6. Summary 039 split up in clauses.

4. Conclusion

This report describes the XML (semi) compliant codification scheme used in ResDial project. It shows how each corpus in this project is stored within directories, along with the way that information relevant to each annotation stage must be included in the files. Even though the most of the project complies with the XML model, some requirements, such as the existence of overlapping and embedding, for example, make some files astray from this model.

Designed to allow for stand-off annotation, each corpus in the project comprises at least three files: one file with the plain text, one with its segmentation in basic units of annotation, and one with the classification of each unit according to some specific annotation scheme. Naturally, both segmentation and classification tasks admit the inclusion of multiple files to the same source text (as a product of different annotators), thereby allowing for their comparison and execution of agreement analyses in the corpora.

This standardisation of ResDial corpora codification, in turn, serves not only the purpose of making it easier to read and use by other people, but also of allowing for the development of a myriad of tools for this project. Along these lines, it has already been developed the prototype of a segmenter (see [Rodrigues et al. 2012]), there also being other tools currently in development.

```
<?xml version="1.0" encoding="UTF-8"?>
<annotation>
    <info type="id" value="01">
    <info type="scheme" value="A1">
    <info type="annotator" value="ann01">
    <info type="source" value="01">
    <info type="source-corpus" value="S1">
    <mark unit="0400">
        <ann type="CATE" value="Negative report about the
                                vendor">
        <ann type="INTE" value="Low">
        <ann type="CONS" value="-">
    </mark>
    <mark unit="0401">
        <ann type="CATE" value="Negative report about the
                                vendor">
        <ann type="INTE" value="Non-low">
        <ann type="CONS" value="-">
    </mark>
    <mark unit="0402">
        <ann type="CATE" value="Neutral report">
        <ann type="INTE" value="">
        <ann type="CONS" value="">
    </mark>
    <mark unit="0403">
        <ann type="CATE" value="Neutral report">
        <ann type="INTE" value="">
        <ann type="CONS" value="">
    </mark>
    <mark unit="0404">
        <ann type="CATE" value="Neutral report">
        <ann type="INTE" value="">
        <ann type="CONS" value="">
    </mark>
    <mark unit="0405">
        <ann type="CATE" value="Neutral report">
        <ann type="INTE" value="">
        <ann type="CONS" value="">
    </mark>
    <mark unit="0406">
        <ann type="CATE" value="Neutral report">
        <ann type="INTE" value="">
        <ann type="CONS" value="">
    </mark>

    (continues on next page ...)
```

Figure 7. Classification of the units in Figure 2.

```
(... continuation)

 <mark unit="0407">
     <ann type="CATE" value="Neutral report">
     <ann type="INTE" value="">
     <ann type="CONS" value="">
</mark>
<mark unit="0408">
     <ann type="CATE" value="Neutral report">
     <ann type="INTE" value="">
     <ann type="CONS" value="">
</mark>
<mark unit="0409">
     <ann type="CATE" value="Neutral report">
     <ann type="INTE" value="">
     <ann type="CONS" value="">
</mark>
<mark unit="0410">
     <ann type="CATE" value="Neutral report">
     <ann type="INTE" value="">
     <ann type="CONS" value="">
</mark>
<mark unit="0411">
     <ann type="CATE" value="Neutral report">
     <ann type="INTE" value="">
     <ann type="CONS" value="">
</mark>
<mark unit="0412">
     <ann type="CATE" value="Neutral report">
     <ann type="INTE" value="">
     <ann type="CONS" value="">
</mark>
<mark unit="0413">
     <ann type="CATE" value="Negative report about the
                             vendor">
     <ann type="INTE" value="Non-low">
     <ann type="CONS" value="-">
</mark>
<mark unit="0414">
     <ann type="CATE" value="Neutral report">
     <ann type="INTE" value="">
     <ann type="CONS" value="">
</mark>
</annotation>
```

Figure 8. Classification of the units in Figure 2 (cont.).

```
<?xml version="1.0" encoding="UTF-8"?>
<participant>
    <info type="id" value="ann01">
    <info type="target-corpus" value="A1">
    <info type="gender" value="m">
    <info type="age" value="20-30">
    <info type="study" value="mphil student">
    <info type="area" value="exact sciences">
</participant>
```

Figure 9. Characterisation of the annotator in Figure 7.

References

Ide, N. and Brew, C. (2000). Requirements, tools, and architectures for annotated corpora. In *Proceedings of Data Architectures and Software Support for Large Corpora*, pages 1–5, Paris, France. European Language Resources Association.

Krauthammer, M., Johnson, S. B., Hripcsak, G., Campbell, D. A., and Friedman, C. (2002). Representing nested semantic information in a linear string of text using xml. In *Proceedings of the AMIA 2002 Symposium*, pages 405–409, San Antonio, TX, USA. ISBN 1-56053-600-4.

O'Donnell, M. (2008). The uam corpustool: software for corpus annotation and exploration. In *Proceedings of the XXVI Congreso de AESLA*, Almeria, Spain.

Ogren, P. V. (2006). Knowtator: A plug-in for creating training and evaluation data sets for biomedical natural language systems. In *Proceedings of the 9th International Protégé Conference*, Stanford, USA.

Orăsan, C. (2003). Palinka: A highly customisable tool for discourse annotation. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialog*, pages 39–43, Sapporo, Japan.

Reidsma, D., sa Jovanović, N., and Hofs, D. (2005). Designing annotation tools based on properties of annotation problems. In *Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research.*

Rodrigues, F., Semolini, R., Roman, N. T., and Monteiro, A. M. (2012). Tseg – a text segmenter for corpus annotation. In *Proceedings of the VIII Brazilian Symposium on Information Systems (SBSI 2012)*, São Paulo, SP, Brazil.

Roman, N. T. (2012). Resdial – descrição da codificação (v.1.0). Technical Report 001/2012, PPgSI-EACH-USP, São Paulo, SP – Brazil.

Roman, N. T. and Carvalho, A. M. B. R. (2010). A multi-dimensional annotation scheme for behaviour in dialogues. In Kuri-Morales, A. and Simari, G. R., editors , *Proceedings of the 12th Ibero-American Conference on Artificial Intelligence (IBERAMIA 2010)*, volume 6433 of *Advances in Artificial Intelligence*, pages 386–395, Bahía Blanca, Argentina. Springer. ISBN: 978-3-642-16951-9.

Roman, N. T., Piwek, P., and Carvalho, A. M. B. R. (2006a). *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, chapter Politeness and Bias in Dialogue Summarization: Two Exploratory Studies, pages 171–185. Springer Netherlands, Dordrecht, The Netherlands. ISBN: 1-4020-4026-1.

Roman, N. T., Piwek, P., and Carvalho, A. M. B. R. (2006b). A web-experiment on dialogue classification. In Rezende, S. O. and da Silva Filho, A. C. R., editors , *Pro-*

ceedings of the Fourth Workshop in Information and Human Language Technology (TIL'2006), Ribeir ao Preto, Brazil. ICMC-USP.

van der Vliet, N., Berzlánovich, I., Bouma, G., Egg, M., and Redeker, G. (2011). Building a discourse-annotated dutch text corpus. *Bochumer Linguistische Arbeitsberichte*, 3:157–171. ISSN: 2190-0949.

Verhagen, M. (2010). The brandeis annotation tool. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 3638–3643, Valletta, Malta.

A. Tagset used in ResDial

Tag	*Where used*	*Usage*
ann	Annotated text	Defines the classification of an annotation unit according to some predefined category. It has the parameters “type”, indicating the category’s name and “value”, corresponding to its value. Must be used within mark.
annotation	Annotated text	Defines the limits of the annotated text. It has a corresponding </annotation>
document	Segmented text	Defines the limits of the segmented text. It has a corresponding </document>
info	All documents	Presents additional information. It has the parameters “type”, used to define the name of the information to be added, and “value”, which holds that information
mark	Annotated text	Defines the application of the annotation scheme to an annotation unit. It has the parameter “unit”, which holds the annotated unit’s identifier. Has a corresponding </mark>
participant	Participant	Defines the limits to the participant characterisation data. It has a corresponding </participant>
plainDocument	Plain text	Defines the limits of the plain text. It has a corresponding </plainDocument>
text	Plain text	Defines the document text (UTF-8 encoded). It has a corresponding </text>
unit	Segmented text	Defines a minimum unit of annotation. It has the parameter “id”, which holds the unit’s unique identifier. It has a corresponding </unit>. “id” must be both in <unit> and </unit>
unit ind	Segmented text	Defines an independent unit. It has the parameter “id”, which holds the unit’s unique identifier. It has a corresponding </unit ind>. “id” must be both in <unit ind> and </unit ind>
<!--	All documents	Defines a comment. It has a corresponding -->