The Methodius Corpus of Rhetorical Discourse Structures and Generated Texts

Amy Isard
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
amy.isard@ed.ac.uk

Abstract

Using the Methodius Natural Language Generation (NLG) System, we have created a corpus which consists of a collection of generated texts which describe ancient Greek artefacts. Each text is linked to two representations created as part of the NLG process. The first is a content plan, which uses rhetorical relations to describe the high-level discourse structure of the text, and the second is a logical form describing the syntactic structure, which is sent to the OpenCCG surface realization module to produce the final text output. In recent work, White and Howcroft (2015) have used the SPaRKy restaurant corpus, which contains similar combination of texts and representations, for their research on the induction of rules for the combination of clauses. In the first instance this corpus will be used to test their algorithms on an additional domain, and extend their work to include the learning of referring expression generation rules. As far as we know, the SPaRKy restaurant corpus is the only existing corpus of this type, and we hope that the creation of this new corpus in a different domain will provide a useful resource to the Natural Language Generation community.

Keywords: Corpus, Natural Language Generation, Discourse Structure

1. Introduction

Most rule-based Natural Language Generation (NLG) systems use a pipeline of hand-crafted components to generate texts (Reiter and Dale, 2000), requiring expert knowledge (see Section 2). There has been recent progress on using machine learning for content planning (Barzilay and Lapata, 2008; Duboue and McKeown, 2001), broad coverage surface realization (Rajkumar and White, 2014), and concept-to-text systems (Konstas and Lapata, 2013; Konstas, 2014), but these do not produce complicated discourse relations. Angeli et al. (2010) and Konstas and Lapata (2013) have researched machine learning methods to create end-to-end NLG systems, but these create more limited discourse structures than those produced by the traditional pipeline systems. To begin to fill this gap, recent research (White and Howcroft, 2015) has attempted to learn rules for the combination of clauses, using the SPaRKy Restaurant Corpus (Walker et al., 2004; Walker et al., 2007).

SPaRKy (Sentence Planning with Rhetorical Knowledge) is a sentence planner which uses rhetorical relations and adapts to a user's individual sentence planning preferences. The SPaRKy Restaurant Corpus pairs text plans which contain rhetorical structure with the resulting restaurant recommendation texts. White and Howcroft (2015) obtain OpenCCG logical forms (see Section 2.3) by parsing the texts, and their rule-induction algorithm learns aggregation and discourse connective rules. These automatically-derived discourse rules could be used in an NLG system in place of the hand-written rules which normally form part of the generation process.

White and Howcroft (2015, p.29) note that "To our knowledge, the SRC remains the only publicly available corpus of input-output pairs for an NLG system using discourse structures with rhetorical relations." We have therefore generated a corpus of texts from the Methodius Natural Language Generation System (see Section 3), using information about a collection of Greek artefacts from the M-PIRO project (Isard et al., 2003), so that their clause-combining induction algorithms can be tested on another domain. The Methodius corpus will be freely available through LREC, and we hope that this rich resource will also prove useful to many other researchers, as the SPaRKy restaurant corpus has. Another possible use of the corpus would be in using the sentence plans and the generated texts to learn rules for referring expression generation.

We present a summary of some Natural Language Generation components in Section 2, and Rhetorical Structure in Section 2.2. The Methodius NLG system is described in Section 3, and the structure of the corpus is described in Section 4, along with an example output.

2. Natural Language Generation

2.1. NLG Pipeline Architecture

The typical phases of the NLG pipeline can be described as follows:

content selection: the choice of information to be presented, based on communicative goals, user models, and dialogue or discourse history,

sentence planning: the structure of the text, and how the pieces of information will be combined into clauses and sentences — this stage can include aggregation and referring expression generation,

surface realisation: creation of the final output text, according to syntactic and morphological rules.

The output of the content selection stage can be expressed in terms of Rhetorical Structure Theory (see Section 2.2 below), and the sentence plans equate to OpenCCG logical forms (see Section 3). Both of these appear in the Methodius corpus, along with the final generated text, as described in Section 4.

2.2. Rhetorical Structure Theory

Rhetorical Structure Theory (Mann and Thompson, 1998) describes a set of relations which exist between proposi-

tions in a discourse. These are classified as either mononuclear relations, which connect a dominant textnode (the 'Nucleus') to its dependent (the 'Satellite'), or multinuclear relations which connect textnodes of equal status. Appropriate rhetorical relations vary according to the domain under consideration, and reflect the structure of the texts. RST has been applied in a number of Natural Language Processing domains, including parsing, summarization, argument evaluation, machine translation, and most frequently NLG (Bontcheva, 2005). The particular rhetorical relations used by Methodius are described in Section 3.

2.3. OpenCCG

OpenCCG is a collection of natural language processing tools, which provide parsing and realisation support based on the CCG grammar formalism (White, 2006; White and Rajkumar, 2009). It takes as input a a lexicon and a set of grammar rules, and a logical form which describes the structure of a particular sentence to be generated. A wide-coverage grammar of English is available for OpenCCG (White et al., 2007), or the grammar and lexicon can be hand-authored for a specific domain, allowing for more efficient processing.

3. Methodius

The Methodius Natural Language Generation System takes a database containing information about entities, for example objects in a museum, and outputs personalized descriptions of the objects such as the one shown in Figure 2. The system can be used in a virtual museum setting, where a visitor clicks on pictures of exhibits, or in a real museum, on a handheld device which a visitor carries with them. The software keeps track of the visitor's path through the collection, so it avoids repeating information, and makes comparisons with previously seen objects (Isard, 2007; Marge et al., 2008).

Methodius was based on the M-PIRO Exprimo system (Isard et al., 2003; Melengoglou et al., 2002) which in turn was based on the ILEX Intelligent Labelling Explorer (O'Donnell et al., 2001; Knott et al., 2001). In common with these previous systems, it uses a pipeline architecture (see Section 2.1), where the first step is content selection, followed by sentence planning, and then realization.

In Methodius, the content selection is based on interest values for each attribute of each object (which for the M-PIRO project data hand-authored by domain experts), and also tries to include at least one comparison with a previous exhibit, if one is available (Isard, 2007). After content selection the text planning component combines the propositions using a variety of aggregation strategies, and builds a syntactic/semantic logical form which is sent to the OpenCCG realizer (see Section 2.3). The M-PIRO domain OpenCCG grammar was hand-authored during the project, and will also be made available as part of the corpus. When working with the SPaRKy corpus, White and Howcroft (2015) parsed the generated texts with the OpenCCG wide coverage grammar; they plan to do the same with the Methodius texts, and the hand-authored grammar may provide a useful counterpart.

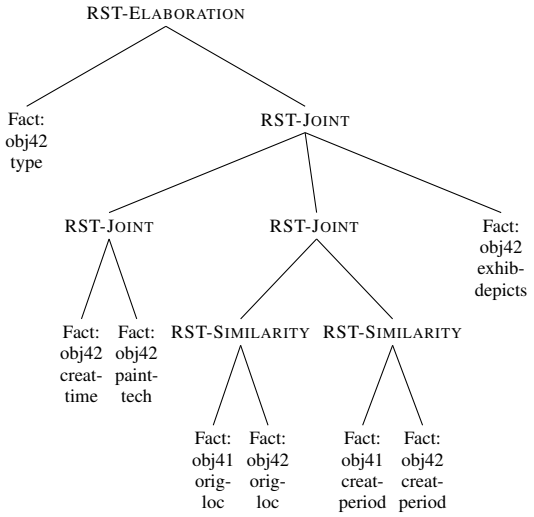

Figure 1: Methodius RST Tree

The rhetorical relations used by Methodius are very similar to those in Exprimo, which are described in detail in (Melengoglou, 2002; Melengoglou et al., 2002).[1] The relations are described below:

RST-ELABORATION - a mononuclear relation, where the nucleus refers to an object and the satellite introduces an attribute of the object, e.g. *This exhibit is an amphora and it was created during the archaic period.*

RST-JOINT - a multinuclear relation which simply connects two facts of equal status, e.g. *This amphora was created during the archaic period and it was decorated using the red-figure technique.*

RST-SIMILARITY - a multinuclear relation which compares two matching attributes of different objects, e.g. *Like the last vessel you saw, this amphora was originally from Attica.*

RST-CONTRAST - a multinuclear relation which compares two non-matching attributes of different objects, e.g. *Unlike the previous coins you saw, which are located in the Athens Numismatic Museum, this tetradrachm is located in the National Museum of Athens.*

An example RST tree is shown in Figure 1, which corresponds to the XML representation of the RST shown in Figure 3, and generates the text in Figure 2. The next section contains a detailed description of the corpus structure.

4. Corpus

The Methodius corpus consists of a set of files each of which contains content plans, logical forms and descriptions for ten exhibits from the M-PIRO database. The M-PIRO database contains 50 exhibits, each of which has between four and nine facts associated with it, and many of

[1] This set of relations overlaps with those used in the SPaRKy corpus, but because of differences in the domain and the nature of the propositions, Methodius includes RST-SIMILARITY and omits RST-JUSTIFY.

This is another lekythos; it was created in between 470 and 460 B.C. and it was decorated with the red figure technique. Like the previous vessel you saw, this lekythos was originally from Attica and it was created during the classical period. It shows an athlete preparing to throw his javelin.

Figure 2: Methodius Generated Text

```
<rst type="elaboration">
 <fact pred="type" arg1="obj42" compare="additive"/>
 <rst type="joint">
  <rst type="joint">
   <fact predicate="creation-time"
         arg1="obj42"
         arg2="obj42-time"/>
   <fact pred="painting-technique"
         arg1="obj42"
         arg2="red-fig-technique"/>
  </rst>
  <rst type="joint"
   <rst type="similarity">
    <fact predicate="original-loc"
          arg1="obj41"
          arg2="attica"/>
    <fact predicate="original-loc"
          arg1="obj42"
          arg2="attica"/>
   </rst>
   <rst type="similarity">
    <fact predicate="creation-period"
          arg1="obj41"
          arg2="classical-period"/>
    <fact predicate="creation-period"
          arg1="obj42"
          arg2="classical-period"/>
   </rst>
   <rst>
    <fact predicate="exhibit-depicts"
          arg1="obj42"
          arg2="obj42-depicts"/>
  </rst>
</rst>
```

Figure 3: Methodius Rhetorical Structure Content Plan

these facts have additional dependent facts, as described in (O'Donnell et al., 2001). For example, one exhibit is a portrait of Alexander the Great, and the database also contains some biographical information about Alexander. The 50 exhibits in the domain therefore provide too many potential paths to generate an exhaustive corpus, so we have chosen to generate sets of texts from exhibit chains, setting the length of the chains to 10 and the number of facts generated per exhibit to 6. The generation output is adaptive given the history of objects already described, so a given exhibit will have varying descriptions depending on its position in the chain and the specific exhibits which precede it. We have randomly selected the exhibits for each chain from the 50 exhibits, allowing the same exhibit to appear more than once in a chain. We have generated 500 chains (5000 texts) for the first version of the corpus.

4.1. Example Description

We include an example of a description of a single exhibit, identified in the database as `obj42`. In this case, the exhibit is the second one in a chain, and the first object in the chain was `obj41`. Both objects are of type lekythos, and

```
<lf>
 <node pred=';' id='id0' mood='dcl'>
  <rel name='Arg1'>
   <node id='id1' pred='be-verb' tense='pres'
         voice='active'>
    <rel name='ArgOne'>
     <node id='id2' pred='this' num='sg'/>
    </rel>
    <rel name='ArgTwo'>
     <node id='id3' pred='lekythos' det='another'
           num='sg'/>
    </rel>
   </node>
  </rel>
  <rel name='Arg2'>
   <node pred='and' id='id4'>
    <rel name='Arg1'>
     <node id='id5' pred='create' tense='past'
           voice='passive'>
      <rel name='ArgOne'>
       <node id='id6' pred='pro3n' num='sg'/>
      </rel>
      <rel name='ArgTwo'>
       <node id='id7' pred='obj42-creation-time'
             prep='in'/>
      </rel>
     </node>
    </rel>
    <rel name='Arg2'>
     <node id='id8' pred='decorate' tense='past'
           voice='passive'>
      <rel name='ArgOne'>
       <node id='id9' pred='pro3n' num='sg'/>
      </rel>
      <rel name='ArgTwo'>
       <node id='id10' pred='red-fig-technique'
             num='sg' prep='with'/>
      </rel>
     </node>
    </rel>
   </node>
  </rel>
 </node>
</lf>
```

Figure 4: OpenCCG Logical Form for First Generated Sentence

they have some attributes in common — namely that they had the same original location and creation period. The content planning module of the Methodius system chooses the most interesting facts about `obj42` based on the possible comparisons with `obj41` and the interest values for the facts in the database. A content plan is created, shown in Figure 3, which describes the same rhetorical relations as the tree in Figure 1. Next, Methodius carries out sentence planning, deciding how to combine multiple facts into sentences, and choosing the most appropriate referring expression for each object given the discourse history. OpenCCG logical forms (Figures 4, 5 and 6) are created which express the sentence planning, and these are sent to the OpenCCG realizer, which generates the output text shown in Figure 2.

5. Conclusions

We have created the Methodius Corpus, which provides a set of generated texts linked to their content plans (with rhetorical relations), and sentence plans (OpenCCG logical forms). The corpus will be used as a second test domain for the induction of clause-combination rules described in White and Howcroft (2015) and can also serve as a resource for the learning of referring expression strategies and other future research efforts which aim to use machine learning techniques to automate parts of the Natural Language Generation process.

```
<lf>
 <node id="id0" pred="like" mood="dcl">
  <rel name="Comparator">
   <node id="id1" pred="vessel" det="def"
           num="sg">
    <rel name="RedGenRel">
     <node id="id2" pred="see" tense="past"
          voice="active">
      <rel name="ArgOne">
       <node id="id3" pred="pro2"/>
      </rel>
      <rel name="ArgTwo">
       <node idref="id1"/>
      </rel>
     </node>
    </rel>
    <rel name="HasProp">
     <node id="id4" pred="previous"/>
    </rel>
   </node>
  </rel>
  <rel name="Focus">
   <node id="id5" pred="and">
    <rel name="Arg1">
     <node id="id6" pred="be" tense="past"
          voice="active">
      <rel name="ArgOne">
       <node id="id7" pred="lekythos" det="dem-prox"
           num="sg"/>
      </rel>
      <rel name="ArgTwo">
       <node id="id8" pred="attica" num="sg"
           prep="from"/>
      </rel>
      <rel name="HasProp">
       <node id="id9" pred="originally"/>
      </rel>
     </node>
    </rel>
    <rel name="Arg2">
     <node id="id10" pred="create" tense="past"
          voice="passive">
      <rel name="ArgOne">
       <node id="id11" pred="pro3n" num="sg"/>
      </rel>
      <rel name="ArgTwo">
       <node id="id12" pred="classical-period" num="sg"
           prep="during"/>
      </rel>
     </node>
    </rel>
   </node>
  </rel>
 </node>
</lf>
```

Figure 5: OpenCCG Logical Form for 2nd Generated Sentence

```
<lf>
 <node id="id0" pred="show" mood="dcl" tense="pres"
          voice="active">
  <rel name="ArgOne">
   <node id="id1" pred="pro3n" num="sg"/>
  </rel>
  <rel name="ArgTwo">
   <node id="id2" pred="obj42-exhibit-depicts"/>
  </rel>
 </node>
</lf>
```

Figure 6: OpenCCG Logical Form for 3rd Generated Sentence

6. Acknowledgements

Thanks to Michael White for discussions which led to the creation of this corpus. The M-PIRO project (2000-2003), which collected the artefact data, was funded by the EC grant ST-1999-10982 and the Methodius project, which supported the development of the Exprimo software, was funded by a Scottish Enterprise Proof of Concept Grant.

7. Bibliographical References

Angeli, G., Liang, P., and Klein, D. (2010). A simple domain-independent probabilistic approach to generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Barzilay, R. and Lapata, M. (2008). Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Bontcheva, K. (2005). Generating tailored textual summaries from ontologies. In *The Semantic Web: Research and Applications*, pages 531–545. Springer.

Duboue, P. A. and McKeown, K. R. (2001). Empirically estimating order constraints for content planning in generation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 172–179. Association for Computational Linguistics.

Isard, A., Oberlander, J., Matheson, C., and Androutsopoulos, I. (2003). Speaking the users' languages. *Intelligent Systems, IEEE*, 18(1):40–45.

Isard, A. (2007). Choosing the best comparison under the circumstances. In *Proceedings of the International Workshop on Personalization Enhanced Access to Cultural Heritage (PATCH07)*.

Knott, A., Oberlander, J., ODonnell, M., and Mellish, C. (2001). Beyond elaboration: The interaction of relations and focus in coherent text. *Text representation: linguistic and psycholinguistic aspects*, pages 181–196.

Konstas, I. and Lapata, M. (2013). A Global Model for Concept-to-Text Generation. *J. Artif. Intell. Res.(JAIR)*, 48:305–346.

Konstas, I. (2014). *Joint models for concept-to-text generation*. Ph.D. thesis, University of Edinburgh.

Mann, C., W. and Thompson, A., S. (1998). Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 3:243–281.

Marge, M., Isard, A., and Moore, J. (2008). Creation of a new domain and evaluation of comparison generation in a natural language generation system. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 169–172. Association for Computational Linguistics.

Melengoglou, A., Androutsopoulos, I., Calder, J., Callaway, C., Clark, R., Dimitromanolaki, A., Hughson, I., Isard, A., Matheson, C., Not, E., and others. (2002). Generation Components and Documentation for Prototype D4. 5. *M-PIRO Project (IST-1999-10982) Public Deliverable*, 1.

Melengoglou, A. (2002). Multilingual Aggregation in the M-PIRO System. Master's thesis, University of Edinburgh.

O'Donnell, M., Mellish, C., Oberlander, J., and Knott, A. (2001). ILEX: An Architecture for a Dynamic Hypertext Generation System. *Nat. Lang. Eng.*, 7(3):225–250, September.

Rajkumar, R. and White, M. (2014). Better Surface Realization through Psycholinguistics. *Language and Linguistics Compass*, 8(10):428–448, October.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, U.K.

Walker, M. A., Whittaker, S. J., Stent, A., Maloor, P., Moore, J., Johnston, M., and Vasireddy, G. (2004). Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, 28(5):811–840.

Walker, M. A., Stent, A., Mairesse, F., and Prasad, R. (2007). Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research*, pages 413–456.

White, M. and Howcroft, D. M. (2015). Inducing Clause-Combining Rules: A Case Study with the SPaRKy Restaurant Corpus. *ENLG 2015*, page 28.

White, M. and Rajkumar, R. (2009). Perceptron reranking for CCG realization. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, August.

White, M., Rajkumar, R., and Martin, S. (2007). Towards broad coverage surface realization with ccg. In *Proc. of the Workshop on Using Corpora for NLG: Language Generation and Machine Translation (UCNLG+ MT)*.

White, M. (2006). Efficient realization of coordinate structures in combinatory categorial grammar. *Research on Language and Computation*, 4(1):39–75.