The E2E NLG Challenge: End-to-End Generation through Partial Template Mining

Charese Smiley[1], **Elnaz Davoodi** [2], **Dezhao Song**[1], **Frank Schilder**[1]
[1] Research & Development, Thomson Reuters, 610 Opperman Drive, Eagan, MN, USA
[2] Center for Cognitive Computing, Thomson Reuters, 120 Bremner Blvd, Toronto, ON, M5J 3A8, CA
firstname.lastname@thomsonreuters.com

Abstract

This paper provides a system description for one of the primary entries by the Thomson Reuters team to the 2017 E2E Challenge. This system mines training data for partial templates using a rule-based approach. Although, this system did not score above the baseline using automatic metrics, we score in the third highest cluster for quality with human evaluation.

1 Introduction

This paper describes one of the primary systems created by the Thomson Reuters team for the 2017 E2E Shared Task. For this challenge, we entered two primary systems: one rule-based described in this paper and the second a sequence-to-sequence approach outlined in (Davoodi et al., 2018).

The E2E challenge involves taking a meaning representation as input and generating natural language output from it. For this entry, we entered a rule-based system which operates in a similar fashion to previous template based natural language generation (NLG) systems (Kondadadi et al., 2013). This system could be used to generate more training data for downstream applications such as neural network generators or used directly to supply generations to the end user. The data used in this challenge is outlined in Section 2. Methods can be found in Section 3. Sections 4 and 5 detail results and discussion and conclusions are given in Section 6.

2 Data

The data for this task comes from a crowdsourced dataset in the restaurant domain collected using CrowdFlower (Novikova et al., 2016). The dataset consists of 50,602 instances derived from 5,751 unique meaning representations (MRs) (Novikova et al., 2017). The current dataset has the advantage of being larger than previous end-to-end datasets such as BAGEL (Mairesse et al., 2010) and SF Hotels/Restaurants (Wen et al., 2015).

To create the data, crowd workers were asked to create a verbalization based on a given MR. They were allowed to omit information if they did not find it useful. The MR could contain up to 8 different attributes: *name, eat type, food, price range, customer rating, area, family friendly*, and *near* in combinations of between 3 and 8 attributes. In 40% of the instances, verbalizations contain either omissions or additional information. The dataset is split in a 76.5/8.5/15 ratio into training, development, and testing.

A sample MR and natural language instance from the training set is given below:

MR:
name[Alimentum],
area[city centre],
familyFriendly[no]

NL:
There is a place in the city centre, Alimentum, that is not family-friendly.

3 Methods

This section provides an overview of the methods used in this system. Training data is delexicalized as described in Section 3.1. This creates a template dictionary which is then expanded through partial template mining in Section 3.2. Finally, Sections 3.3 and 3.4 discuss sentence planning and realization of the final natural language generation.

3.1 Delexicalization

First, we delexicalized the data using a simple string match to automatically replace the attribute values contained in the MR with the attribute name. We did this for all attributes except for *family friendly* which has a wide range of potential realizations (e.g., positive: *children are wel-*

	Attributes	Template
Original	customer rating, name, eatType, food, near, area	With a rating of _CUSTOMER RATING_, _NAME_ _EATTYPE_ serves _FOOD_ food. It is located near _NEAR_ and _AREA_.
Partial 1	customer rating, name, eatType, food	With a rating of _CUSTOMER RATING_, _NAME_ _EATTYPE_ serves _FOOD_ food.
Partial 2	near, area	It is located near _NEAR_ and _AREA_.

Table 1: Partial templates extracted from training data.

come, kid friendly, or negative: *adult only, not for kids*). Therefore, we use a binary yes/no value for that attribute. For each delexicalized sentence, we check to see whether all attributes in the MR were captured during the delexicalization process. If there is a difference between the number of attributes in the MR and the number that were successfully delexicalized, we discard that instance. In total, we discarded roughly 45% of the training sentences. This is slightly more than the 40% of instances in the data that contained omissions or additions. We then use the delexicalized templates to create a dictionary look-up of the MRs.

A sample delexicalized template would be:

There is a place in _AREA_, _NAME_, that is _NO.

3.2 Partial Template Mining

With the templates now identified, we identify templates that are composed of multiple sentences and split along sentence boundaries. The individual sentences are then stored as partial templates along with the attributes reverse engineered from the templates. Table 1 shows the original template containing 2 sentences and the derived partial templates containing one sentence each. Through this process we collect templates containing all 8 of the attributes individually as well as combinations from 2-8. By extracting individual templates for each attribute alone, we guarantee that we can cover any combination of attributes by generating up to 8 separate sentences although this would not sound very natural.

3.3 Sentence Planning

In the testing phase, we are supplied with an MR which may consist of an unseen combination of attributes. We treat the attributes of the MR as a set filling the templates using the following algorithm:

1. Look up the attribute set from current MR in the template dictionary. There are 23,203 attribute sets with an average of 140 templates per attribute set. If the attribute set exists, take the set of candidate templates with those attributes (skip to Step 5).

2. If the attribute set does not exist in the dictionary, generate the partitions of the set creating subsets where every element of the set appears in one and only one subset. For example, attributes could be partitioned as ({*customer rating, name,eatType,food,near,area*}),({*customer rating*}, {*name,eatType,food,near,area*}), and so on where each subset in a partition corresponds to a partial template. The subsets are ordered so that the first subset corresponds to the first sentence in the NL generation and so on. However, the elements in the subset are unordered. After this step there are an additional 5,763 attribute sets with an average of 30 subsets (partial templates) per attribute set.

3. Randomly select one partition. If the second partition in Step 2 is selected, for example, there will be 647 partial templates for {*customer rating*} and 117 for {*name,eatType,food,near,area*}.

4. If a subset contains the restaurant *name* attribute, promote the entire subset to the front to avoid generating cataphoric utterances. Thus, ({*customer rating*}, {*name,eatType,food,near,area*}) becomes ({*name,eatType,food,near,area*},{*customer rating*}), for example.

5. Relexicalize all of the (full or partial) templates from the attribute set using the unaltered entities from the MR.

6. Perform basic rule-based clean-up (e.g., convert *an* to *a* when the following word consists of a consonant).

	BLEU	NIST	METEOR	ROUGE_L	CIDEr	Quality	Naturalness
BASELINE	0.6593	8.6094	0.4483	0.6850	2.2338	-	-
SYSTEM1	0.4111	6.7541	0.3970	0.5435	1.4096	-	-
SYSTEM2	0.4202	6.7686	0.3968	0.5481	1.4389	3 of 5	5 of 5

Table 2: Results of E2E automatic and human evaluations.

3.4 Realization

All templates in the candidate set are relexicalized with the current MR. From there we filter candidates by performing basic sentiment analysis using NLTK's sentiment analyzer[1] and removing sentences whose sentiment is incongruent (e.g., great restaurant described as having low rating). To determine this, we look for sentences with non-neutral scores for both positive and negative polarities but no word indicating a reversal such as *however*. The final output from the candidate set is selected at random.

4 Results

Evaluation for the E2E was conducted using both automatic metrics and human scoring. These results are given in Table 2 with the automatic scoring described in Section 4.1 and the human evaluation in Section 4.2.

4.1 Automatic Scoring

Table 2 shows the results comparing the baseline system with the results from our system. Systems were evaluated automatically using BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Denkowski and Lavie, 2014), ROUGE_L (Lin, 2004), and CIDEr (Vedantam et al., 2015). The first row contains the results for the BASELINE system – a sequence-to-sequence model with attention (Dušek et al., 2018). The other two rows in the table contain the automatic scores for our system where the results for SYSTEM1 are created from the training data alone and SYSTEM2 is composed of both the training and development data. SYSTEM2 was submitted as the primary system for the rule-based approach because it contained more training data. Here we see that for the automatic metrics, neither system outperforms the baseline system. The addition of the development data in the SYSTEM2 system produces a slight boost in all automatic metrics except for METEOR.

[1] http:\www.nltk.org

4.2 Human Evaluation

For the human evaluation metric (Dušek et al., 2018), raters were shown the reference sentence along with 5 generations from various competing systems. They were asked to rank the generations for *quality* and *naturalness*. For the *quality* ranking, raters were given the MR along with the system reference output. They were asked to rank the output based on grammatical correctness, fluency, adequacy, and so on. *Naturalness* measures whether the utterance could have been written by a native speaker. Raters were not given the MR for the *naturalness* evaluation. Thus, this metric does not take into account faithfulness to the MR. The results of the human evaluation are based on the system's inferred TrueSkill score (Sakaguchi et al., 2014) which is computed based on pairwise comparisons between systems. Human evaluations were only performed on the primary systems submitted by each team. For quality, SYSTEM2 ranked 3rd of 5 clusters of systems. For naturalness, our system was in the 5th of 5 clusters. Systems within each cluster are considered statistically indistinguishable.

5 Discussion

Overall, we find that our rule-based system did not perform well when evaluated on the automatic metrics. This is not particularly surprising as we did not make any attempts to optimize for these metrics.

For the human evaluations, we scored in the third cluster of groups. For a production NLG system, quality of generations would be the most important metric as users would expect faithfulness to the underlying data along with other standards such as grammaticality. Because a variety of features are encompassed within the metric of *quality* (e.g., fluency and adequacy), it is difficult to determine where our algorithm succeeds and fails. To help determine this, we took a closer look at 25 sentences from the test generations. Of those, we found that every generation contained all of the attributes from the MR while one contained ad-

#	MR	NL
1	name[The Waterman], eatType[restaurant], food[Indian], priceRange[moderate], area[city centre], familyFriendly[yes], near[Raja Indian Cuisine]	*The Waterman is a child friendly restaurant which serves moderately priced Indian food. It is near Raja Indian Cuisine in the city centre area.*
2	name[The Wrestlers], eatType[restaurant], food[Italian], priceRange[moderate], area[city centre], familyFriendly[yes], near[Raja Indian Cuisine]	*The Wrestlers is children friendly and serves Italian food. It is near Raja Indian Cuisine in the city centre. It's a restaurant with a moderate price range.*
3	name[The Plough], eatType[pub], food[Chinese], priceRange[high], area[riverside], familyFriendly[yes], near[Raja Indian Cuisine],	*The Plough, a Chinese pub near Raja Indian Cuisine, offers family friendly riverside dining for high.*

Table 3: Sample natural language output of MRs from the test set.

ditional information. However, we noticed a few issues with fluency that may have contributed to lowering the scores for *quality*.

Table 3 shows sample output from our system for a given MR. NL (1) is an example of a good generation in terms of grammaticality, adequacy, and fluency. NL (2) shows a generation that is faithful to the MR, but perhaps is less fluent due to being composed of multiple sentences as a result of our strategy of randomly selecting a partition of attributes that satisfies the MR. Prioritizing partitions that encompass more attributes may be a simple solution. Finally, in NL (3) we see a generation that sounds disfluent due to the insertion of the adjective *high* from the MR where a noun phrase such as *a high price* would have sounded more natural.

6 Conclusion

We described one of the Thomson Reuters' systems entered into the 2017 E2E challenge. This implementation used a rule-based approach to end-to-end natural language generation. Although our system did not score well by automatic metrics, it was able to deliver sentences which are faithful to their underlying MR. In the future, this system can be used as an engine to generate additional training data for statistical approaches.

References

Elnaz Davoodi, Charese Smiley, Dezhao Song, and Frank Schilder. 2018. The E2E NLG Challenge: Training a Sequence to Sequence Approach for Meaning Representation to Natural Language Sentences. In Ondrej Dušek, Jekaterina Novikova, and Verena Rieser, editors, *(in prep.)*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*. pages 376–380.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., pages 138–145.

Ondrej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *(in prep.)*.

Ravi Kondadadi, Blake Howald, and Frank Schilder. 2013. A Statistical NLG Framework for Aggregated Planning and Realization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 1406–1415.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out* .

François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1552–1561.

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th*

Annual Meeting of the Special Interest Group on Discourse and Dialogue. Saarbrücken, Germany. ArXiv:1706.09254.

Jekaterina Novikova, Oliver Lemon, and Verena Rieser. 2016. Crowd-sourcing nlg data: Pictures elicit better data. *arXiv preprint arXiv:1608.00339* .

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 311–318.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *WMT@ ACL*. pages 1–11.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 4566–4575.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745* .