A Statistical Framework for Product Description Generation

Jinpeng Wang[1], **Yutai Hou**[2*], **Jing Liu**[1], **Yunbo Cao**[3] and **Chin-Yew Lin**[1]
[1] Microsoft Research Asia, {jinpwa,liudani,cyl}@microsoft.com
[2] Harbin University of Technology, ythou@ir.hit.edu.cn
[3] Tencent Corporation, Beijing, yunbocao@tencent.com

Abstract

We present in this paper a statistical framework that generates accurate and fluent product description from product attributes. Specifically, after extracting templates and learning writing knowledge from attribute-description parallel data, we use the learned knowledge to decide *what to say* and *how to say* for product description generation. To evaluate accuracy and fluency for the generated descriptions, in addition to BLEU and Recall, we propose to measure what to say (in terms of attribute coverage) and to measure how to say (by attribute-specified generation) separately. Experimental results show that our framework is effective.

Attribute Name	**Attribute Value**
Processor	Intel Core i3-2350M
RAM Size	6 GB
Series	Dell Inspiron
……	
Screen Size	15.6 inches
Hard Drive Size	500 GB

(a) Product attributes

With 6 GB of memory and a Genuine Intel Core i3-2350M processor, this Dell Inspiron laptop will boost your productivity and enhance your entertainment. The bright, 15.6 inches display showcases movies and games in stunning cinema clarity. …

(b) Generated description

Figure 1: Example of generating product description from product attributes.

1 Introduction

In this paper, we study the problem of product description generation, i.e., given attributes of a product, a system automatically generates corresponding description for this product (see Fig. 1). One application for this task is in (voice) QA systems like Amazon Echo, where reading out the attributes of a product is not desirable. We also found that only 45% of descriptions contain more than 50 words after analyzing 40 million products from Amazon. Generating descriptions for the products which do not have descriptions, and explaining complex attributes of the product for better understanding are also valuable.

Data-to-text generation renders structured records into natural language (Reiter and Dale, 2000), which is similar to this problem. Statistical approaches were employed to reduce extensive development time by learning rules from historical data (Langkilde and Knight, 1998; Liang et al., 2009). Duboue and McKeown (2003) proposed a statistical approach to mine content selection rules for biography descriptions; Kondadadi et al. (2013) and Howald et al. (2013) proposed a statistical approach to select appropriate templates for weather report generation.

However, product description generation is different from above work. To generating a useful product description, a system needs to be aware of the relative importance among the attributes of a product and to maintain accuracy at the same time. Successful product description generation needs to address two major challenges: (1) *What to say*: decide which attributes should be included

[*] This work was done when the second author was an intern at Microsoft Research Asia.

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 187–192,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

in the description; (2) *How to say*: decide how to order selected attributes in the description.

To tackle these problems, we introduce a statistical framework. Our approach has three significant merits. (1) *Coherent with fact*: we proposed to learn structured knowledge from training dataset, and use it to choose important attributes and determine the structure of description; (2) *Fluent*: the proposed approach is template-based which guarantees grammaticality of generated descriptions, and the proposed templated knowledge help to choose semantically correct template; (3) *Highly automated*: the proposed approach required only weak human intervention.

Moreover, in addition to the standard metrics for data-to-text generation, e.g, BLEU (Konstas and Lapata, 2013; Lebret et al., 2016; Kiddon et al., 2016); to evaluate accuracy and fluency of generated descriptions, we propose to measure what to say and how to say separately.

2 Problem Definition

Fig. 2 shows the system framework of product description generation. Our system first extracts sentence level templates and learns writing knowledge from a given parallel dataset, then generates a new description for an input data at the online stage by combining sentence level templates using the learned writing knowledge. The latter step which generates document from sentences is the core component of the product description generate framework. It is called *Document Planning* and is our focus in this paper.

Document Planning as a Ranking Problem In the online stage, given the attributes of a product and the extracted templates, we first generate candidate descriptions by combining all valid templates which fit the given attributes, and then rank the candidate descriptions with the learned writing knowledge. After formulating it as a ranking problem, it is flexible to integrate all kinds of features to estimate the quality of the generated descriptions.

Sentence Level Template Extraction Given a parallel dataset, we first align descriptions and theirs corresponding attributes to extract templates. Several studies (Liang et al., 2009; Kondadadi et al., 2013; Lebret et al., 2016) can be applied to solve this problem. In this paper, we follow the approach which is proposed by Kondadadi et al. (2013). Table 1 shows some sample extracted templates.

Original Text:

- The massive 8 GB of memory will allow you to have lots of files open at the same time.
- The D520 laptop installed with Windows 7.

Extracted Sentence Level Templates:

- The massive **[RAM Size]** of memory will allow you to have lots of files open at the same time.
- The <u>**D520**</u> laptop installed with [**Operating System**].

Table 1: Extracted template examples. Words in bracket are aligned attributes; words with underline are attributes missing in template extraction.

3 Document Planning with Writing Knowledge

Product description generation is far more than simply combining sentences level templates. As we have discussed in the introduction, there are two main challenges for this problem: *what to say* and *how to say*. To solve these problems, we propose to learn templated knowledge and structured knowledge, and use them for ranking generated candidate descriptions.

3.1 Templated Knowledge

At the first step of generating description in the online stage, we fill the extracted templates with the attributes of the input data. However, the extracted templates are with different quality or might have semantic gap with the filled values.

Value Preference For the first extracted template shown in Table 1, the context words in this template depend on value of “RAM Size” strongly. This template is more coherent with products whose “RAM Size” is “8 GB” or “16 GB” rather than that is “1 GB”. To calculate the relatedness between attribute value v_a and template t, we define value preference as:

$$\text{ValPref}(v_a, t) = \sum_{v_i \in \text{V}(t)} \Big(1 - \text{Dist}(v_a, v_i)\Big) P(v_i), \quad (1)$$

where $\text{V}(t)$ is all values of an attribute which are extracted from template t in training data[1],

[1] To avoid sparseness on values, we use context words which surrounding attribute to represent template instead of using all words. In this paper, we combine the proceeding two words and the following ten words as context.

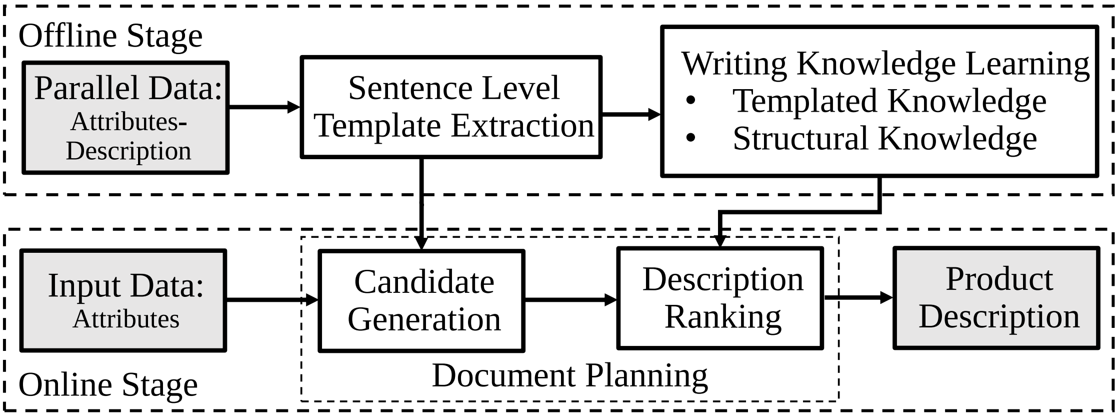

Figure 2: The system framework.

$P(v_i)$ is the probability of value v_i in V(t), and Dist(v_a, v_i) is defined as distance between two values. We can treat all values of attribute as string type, and use normalized editing distance to measure Dist(v_a, v_i)[2]. To improve accuracy for specific domain, for attributes with numerical values[3], Dist$(v_a, v_i) = \frac{|v_a - v_i|}{v_a^{(max)} - v_a^{(min)}}$, where $v_a^{(max)}$, $v_a^{(min)}$ are the upper and lower bound of attribute a in training data.

Missing Attribute For the second extracted template shown in Table 1, "D520" is an attribute value which is missing in template extraction, and such low-quality template with unaligned attribute may hurt the performance of generated description. We define *Missing Attribute* as a word that contains capital letters or numbers, and use this metric as a metric to penalize templates with potential missing attributes during template selection.

3.2 Structured Knowledge

We would also like the generated description contains the important attributes of a product and coherent in semantic. This writing knowledge can be learned from training data.

Attribute Prior Not all attributes of a product are equally important and not all of them are mentioned in a description with the same probability. To capture this information, we define the prior of an attribute a_i as $P(a_i) = \frac{\text{Mention}(a_i)}{\sum_j \text{Mention}(a_j)}$, where Mention$(a_i)$ is the number of mention of attribute a_i in the extracted templates.

Attribute Dependency It is worth noting that attributes which are mentioned in a description are interrelated. For example, in descriptions of computers, "CPU" usually mentioned in the first sentence and "RAM Speed" usually follows "RAM Size". To capture such information, the dependency between attribute a_i and a_j can be defined as $P(a_i|a_j) = \frac{\text{Co-occurrence}(a_i, a_j)}{\sum_k \text{Co-occurrence}(a_k, a_j)}$, where Co-occurrence$(a_i, a_j)$ is the count of a_i and a_j mentioned in consecutive sentences[4]. For a document d which is constructed by sentences $(s_1, ..., s_n)$, where each sentence s_i contains a set of attributes $(a_{i,1}, ...a_{i,|s_i|})$. We assume that current sentence s_i depend only on its previous sentence s_{i-1}, and the structured score for document d can be defined as

$$\text{Struct}(d) = \sum_{i=2}^{n} P(s_i|s_{i-1}) = \sum_{i=2}^{n} \frac{P(s_i, s_{i-1})}{\sum_l P(s_{(l)}, s_{i-1})}, \quad (2)$$

where $P(s_i, s_{i-1})$ has multiple choices, e.g., $\sum_{j,k}\{P(a_{i,j}|a_{i-1,k})\}$, $\max_{j,k}\{P(a_{i,j}|a_{i-1,k})\}$ or $\min_{j,k}\{P(a_{i,j}|a_{i-1,k})\}$.

3.3 Ranking the Generated Descriptions

We first generate candidate descriptions for ranking. Given the attributes of a product, we fill the attribute values into templates which have corresponding slots, and treat all the combinations of filled templates as generated candidate descriptions. We then adopt SVM-rank (Joachims, 2002) with linear kernel to rank the candidate descriptions, and treat the top candidate as the answer. Specifically, we use BLEU score between refer-

[2] Normalized by the longer length of v_a and v_i.

[3] Improvement will be made even if just creating Dist(., .) for the common attributes. In our case, only "RAM Size" and "Hard Disk Size" are treated as with numerical values.

[4] For convenience, two padded sentences [Begin] and [End] are inserted to the start and the end of splited sentences.

ence description and generated description as label score and use the features shown in Table 2.

Basic knowledge:
- # words;
- # sentences;
- # mentioned attributes.

Templated knowledge:
- Value preference: is described by Eq. 1. We calculate the sum, max and min of value preferences for all attributes in candidate document, and treat them as separated features;
- # missing attribute: is described in Section 2.

Structured knowledge:
- Attribute prior: is the sum of attribute priors for attributes mentioned in candidate description.
- Attribute dependency: is described by Eq. 2. The structured scores which based on different version of $P(s_i, s_{i-1})$ are treated as separated features;

Table 2: Features of ranking model.

4 Experiments

4.1 Dataset

We collect the dataset, i.e., (description, attributes) pairs, from category "Computers & Tablets" from Amazon.com, and discard products whose description contains less than 100 words or whose attribute list contains less than five attributes. Table 3 shows the statistics[5]. This dataset has been divided into three parts to provide training (70%), validation (10%) and test sets (20%).

Parameter	**Value**
# (description, attribute table) pairs	25,375
Avg. # of words in description	117.4
Avg. # of sentences in description	4.7
Avg. # of attributes in attribute list	21.2

Table 3: Dataset statistics.

4.2 Compared Methods

We compare these methods in experiments: *Basic*, *+Templated*, *+Structured* and *Full* are ranked based on basic features, basic+templated features, basic+structured features and basic+templated+structured features respectively; *WordCount* and *AttriCount* are rankers which sort candidates in the descending order of word count and attribute count respectively; *OracleBLEU* is an oracle ranker which always chooses the top candidate in term of BLEU as the answer (can be seen as the upper bound of ranking).

4.3 Overall Performance

First of all, we show an example of generated description in Table 5. We adopt language similarity BLEU (Papineni et al., 2002) and retrieval accuracies top-K recall[6] as our evaluation metrics, which are widely used in related work. Table 4 shows that both of structured information and template information help improve the overall performance, and the full model achieves the best performance. Notice that the upper bound in term of BLEU is only 31.5, so the above performance is acceptable. For Recall, both Value Preference and Attribute Prior are the most useful features for retrieving the groundtruth.

Method	**BLEU**	**Recall@3**
Basic	13.2 ($\uparrow$0%)	0.409 ($\uparrow$0%)
+Structured	14.0 ($\uparrow$6.1%)	0.795 ($\uparrow$94.2%)
+Templated	15.3 ($\uparrow$15.9%)	0.717 ($\uparrow$75.2%)
Full	**15.4** ($\uparrow$16.7%)	**0.803** ($\uparrow$96.1%)
WordCount	13.1	0.255
AttriCount	11.3	0.155
OracleBLEU	31.5	1.000

Table 4: Overall performance.

4.4 Attribute Coverage

We propose to evaluate attribute coverage which is important for product description generation. For example, a product in our dataset has 21.2 attributes on average, while only 7.6 attributes are mentioned in a reference description on average. Specifically, we extract attributes which mentioned in a reference description, and compare them with those in its corresponding generated descriptions. From Table 6, we can see that by considering structured information *+Structured* is better than *+Templated* in term of attribute coverage while that is on the contrary in term of BLEU.

[5] The dataset will be available at `http://joopoo.github.io`

[6] The recall rate of a groundtruth (in this case the reference description) retrieval given top K candidates.

Original Description	Generated Description
Dell Inspiron 15 5000 15-5547 15.6” Touchscreen LED (In-plane Switching (IPS) Technology) Notebook - Intel Core i7 i7-4510U 2 GHz - Silver - 16 GB RAM - 1 TB HDD - Intel HD Graphics 4400 - Windows 8.1 64-bit (English) - 1920 x 1080 Display - Bluetooth - E i5547-12500sLV (Product Link: www.amazon.com/dp/B00LGNL5S6)	This Dell Inspiron 15-5558 laptop gives you the everyday features you need, at a great value! It runs many services simultaneously and the dynamic combination of the 2 GHz Intel Core i7 processor and 16 GB RAM enables the acute ability to focus on concurrent tasks with minimal performance degradation. It features an elegant design, includes enhancements to the apps you use most, and enables your PC and Windows 8 devices to work together in amazing ways. With flexible technology and service options, wireless connectivity, and Intel processors packed into a compact system, the Dell Inspiron 15-5558 gives you the essential mobility that will get your business going places.

Table 5: An example of generated description.

It is worth noting that *OracleBLEU* which ranks generated descriptions in term of BLEU performs fair. This is because BLEU only takes word overlap into consideration but not attributes. In other words, descriptions that share same words obtain high BLUE scores, although they are talking about different attributes. For example, descriptions about “RAM” and “Hard Disk” may share same words as “massive” or “GB”. From this point, attribute coverage can be seen as complementary to BLEU.

Method	Precision	Recall	F1
Basic	0.610	0.573	0.573
+Structured	0.615	0.590	0.584
+Templated	0.612	0.580	0.577
Full	**0.623**	0.611	**0.597**
WordCount	0.623	0.543	0.557
AttriCount	0.596	**0.621**	0.589
OracleBLEU	0.605	0.592	0.577

Table 6: Performance on attribute coverage.

4.5 Attribute-Specified Generation

After evaluating attribute coverage, we move to evaluate descriptions which are generated with specific attributes. This task can help us to evaluate quality of generating description by avoiding effect due to attribute selection. In another word, we generate a product description with a given subset of attributes which have been mentioned in the reference description instead of given the whole attributes. From Table 7 we can see better performance for all methods as attributes are specified. Our methods still outperform baselines even when part of features are weakened in this setting, e.g., the prior scores in structured feature. This means that the basic and templated features are also helpful for description generation.

Method	BLEU	
Basic	19.9	(↑0%)
+Structured	20.2	(↑1.5%)
+Templated	20.7	(↑2.0%)
Full	**20.8**	(↑4.5%)
WordCount	19.5	
AttriCount	18.8	
OracleBLEU	30.1	

Table 7: Performance on attribute-specified description generation.

4.6 Human Evaluation

In this evaluation, the following factors are evaluated: (1) Fluency; (2) Correctness: how well the generated description fits corresponding attribute values; (3) Completeness: how well the generated description mentions most of main attributes; and 4) Salient Attribute Mention: how well the generated description highlights its salient attributes. We selected 50 random test products, and for each product we used a Likert scale ($\in [1, 5]$) and report averaged ratings among two annotators.

Table 8 shows the results. The *Full* method beats *WordCount* on all metrics which means that the proposed basic, templated and structured information are helpful. Our *Full* method outperforms *Reference* in term of Completeness as the latter tends to mention fewer attributes in descriptions.

Method	Full	WordCount	Reference
Fluency	4.07	3.67	**4.62**
Correctness	4.03	3.74	**4.87**
Completeness	**4.32**	4.13	4.04
Salient Attr.	4.01	3.76	**4.33**

Table 8: Human evaluation results on the generated and reference descriptions. Score $\in [1, 5]$.

5 Conclusions

In this paper, we proposed a statistical framework for product description generation. The proposed structured information and templated information are helpful for deciding what to say and how to say for description generation. In addition, a new evaluation process is proposed to measure the generated descriptions. The experimental results show that our framework is effective in generating accurate and fluent product description.

References

Pablo A. Duboue and Kathleen R. McKeown. 2003. Statistical acquisition of content selection rules for natural language generation. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 121–128, Stroudsburg, PA, USA. Association for Computational Linguistics.

Blake Howald, Ravikumar Kondadadi, and Frank Schilder. 2013. Domain adaptable semantic clustering in statistical NLG. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 143–154, Potsdam, Germany. Association for Computational Linguistics.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, New York, NY, USA. ACM.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 329–339.

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)*, pages 1406–1415, Sofia, Bulgaria. Association for Computational Linguistics.

Ioannis Konstas and Mirella Lapata. 2013. Inducing document plans for concept-to-text generation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1503–1514.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 704–710, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1203–1213.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 91–99, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Natural Language Processing. Cambridge University Press.