Statistical Natural Language Generation from Tabular Non-textual Data

Joy Mahapatra and **Sudip Kumar Naskar** and **Sivaji Bandyopadhyay**
Jadavpur University, India
joymahapatra90@gmail.com, sudip.naskar@cse.jdvu.ac.in,
sbandyopadhyay@cse.jdvu.ac.in

Abstract

Most of the existing natural language generation (NLG) techniques employing statistical methods are typically resource and time intensive. On the other hand, handcrafted rule-based and template-based NLG systems typically require significant human/designer efforts. In this paper, we proposed a statistical NLG technique which does not require any semantic relational knowledge and takes much less time to generate output text. The system can be used in those cases where source non-textual data are in the form of tuple in some tabular dataset. We carried out our experiments on the Prodigy-METEO wind forecasting dataset. For the evaluation purpose, we used both human evaluation and automatic evaluation. From the evaluation results we found that the linguistic quality and correctness of the texts generated by the system are better than many existing NLG systems.

1 Introduction

The aim of a natural language generation (NLG) system is to produce apprehensible natural language text from non-textual data source which could be a table, an image, numerical data or graphical data (Reiter and Dale, 1997). NLG is just the reverse process of the natural language understanding task.

Although many NLG systems have been proposed so far, there are mainly two types of language generation systems: knowledge-intensive systems and knowledge-light systems (Adeyanju, 2012). Knowledge-intensive NLG systems can be categorized mainly into two categories: template-based systems and handcrafted rule-based systems. Knowledge-intensive generation approaches take significant human effort or expert advise for building an NLG system. Some examples of this type of NLG systems are SumTime system (Reiter et al., 2005), FoG system (Goldberg et al., 1994), PLANDOC system (McKeown et al., 1994), etc. On the other hand, knowledge-light NLG systems mostly use statistical methods to generate output text and take less human effort. Being automatic systems, knowledge-light systems mostly employ machine learning and data mining techniques. There are many types of knowledge-light systems; n-gram based NLG (Langkilde and Knight, 1998), neural network based NLG (Sutskever et al., 2011), case based NLG (Pan and Shaw, 2004), etc. However, it has been observed that knowledge-intensive systems typically perform better than knowledge-light systems as per human evaluation (Adeyanju, 2012).

Beside knowledge-intensive and knowledge-light NLG systems, there are also some NLG systems which can be built through semi-automatic techniques. Probabilistic synchronous context free grammar (PSCFG) based NLG system (Belz, 2008) falls into this category of NLG systems.

In this paper we propose a novel, knowledge-light approach based NLG system which converts a tuple of tabular-formed non-textual data into its corresponding natural language text data. Unlike most of the existing NLG systems, our system does not require any human effort or domain expert help. Moreover, the system does not require much time and computer resources (i.e., hardware equipments) for training and generation purpose. Most of the neural network (especially Recurrent Neural Net-

Proceedings of The 9th International Natural Language Generation conference, pages 143–152,
Edinburgh, UK, September 5-8 2016. ©2016 Association for Computational Linguistics

work) based knowledge-light NLG systems demand advanced computer resources and processing time for training. Contrastingly, without taking much human effort and resources, our system is able to generate intelligible and readable text output.

The remainder of the paper is organized as follows. Section 2 briefly presents relevant related work. The proposed NLG system is described in Section 3. Section 4 elaborates the experimental settings, dataset and the corresponding results. Section 5 concludes the paper.

2 Related works

Till date, a number of knowledge-light approach based language generator systems have been proposed and some of them achieved quite good results in the generation task. Two such successful statistical NLG systems are Nitrogen (Knight and Hatzivassiloglou, 1995; Langkilde and Knight, 1998) and Oxygen (Habash, 2000). These two NLG systems are based on statistical sentence realizer. Similarly Halogen (Langkilde and Knight, 1998) represents another statistical language generator which is based on statistical n-gram language model. Oh and Rudnicky (2000) also proposed an NLG system based on statistical language model.

Since its inception, statistical machine translation (Brown et al., 1993; Koehn, 2010) has gained immense popularity and it is the most prominent approach and represents the state-of-the-art in automatic machine translation. The task of NLG can be thought as a machine translation task because of the similarity between their end objectives - converting from one language to another. Langner and Black (2009) proposed an NLG system, Mountain, which modelled the task of NLG as statistical machine translation (SMT). They used the MOSES[1] toolkit (Koehn et al., 2007) for this purpose. Belz and Kow (2009) proposed another SMT based NLG system which made use of the phrase-based SMT (PB-SMT) model (Koehn et al., 2003). The MOSES toolkit offers an efficient implementation of the PB-SMT model. However, the linguistic quality and readability of PB-SMT based NLG systems were not as good as compared to other statistical NLG systems like Nitrogen, Oxygen, etc. (Belz and Kow, 2009).

Some semi-automatic NLG systems had also been proposed. The Probabilistic synchronous context-free grammar (PSCFG) generator (Belz, 2008) represents this category of NLG systems which can be created mostly automatically but requires manual help to certain extent. In synchronous context-free grammar (SCFG) a pair of CFGs are considered, where one CFG of CFG pair is responsible for the meaning representation and the other CFG of the pair is responsible for generating the natural language text output.

Another type of automatic NLG systems makes use of case-based reasoning (CBR) or instance based learning. This type of CBR based NLG systems are based on the concept that similar set of problems will appear in future and the same set of solutions will compensate or solve those problems. SEGUE NLG system (Pan and Shaw, 2004) was partly a CBR based NLG system; SEGUE was built with a mix of CBR approach and rule based policy. Adeyanju (2012) designed a CBR approach based weather forecasting text generation tool CBR-METEO. The advantage of CBR based system is that it takes very little manual help and if the given prior dataset covers almost all types of input instances then CBR based systems perform better.

Recently, some neural network based NLG systems have been proposed. With the advent of recurrent neural network (RNN) based language models (RNNLM) (Mikolov et al., 2010), some RNN based NLG systems have been proposed. An idea of generating text through recurrent neural network based approach with Hessian-free optimization was proposed by (Sutskever et al., 2011). However, this method takes a long training time. An RNN based NLG technique was proposed by (Wen et al., 2015) based on a joint recurrent and convolutional neural network structure. This system was able to train on dialogue act-utterance pairs without any semantic alignments or predefined grammar trees.

Although rule based knowledge-intensive NLG systems take long time and expert knowledge and feedback to be developed, this type of systems most of the times are able to generate high quality natural language text output. For example, SumTime (Reiter et al., 2005) weather forecasting system is essentially a rule based NLG system, however, its

[1] http://www.statmt.org/moses

output text quality was found to be quite better compared to other automatic NLG systems. Another rule based system, probabilistic context free grammar based generator (Belz, 2008) is also able to generate high quality sentences which mostly correlate well with the given corpus text. In PCFG based system, all possible generation grammars are first discovered manually and then probabilities are assigned to those grammars automatically by observing the corpus.

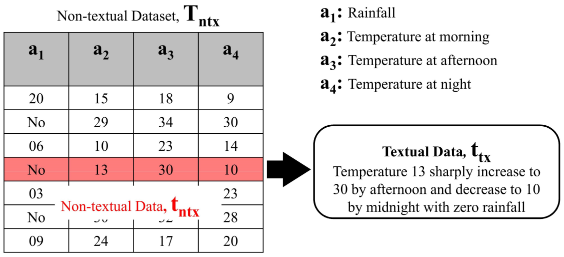
3 System Description

This section describes our knowledge-light, statistical NLG system. Like any other computer software system, the system goes through similar development stages. We describe those stages in the following subsections.

3.1 Task definition

The primary objective of the system is to generate natural language text output from tuple record of a non-textual table structure dataset. The table contains a set of different attributes (qualitative or quantitative). Each row or tuple of the table represents a single unit of non-textual data which represents a vector of that particular tuple's attribute values.

Figure 1 visualizes this task for a typical weather forecasting application. According to that figure, the task is to generate textual data t_{tx} from a tuple-formed non-textual data t_{ntx} of the T_{ntx} dataset (table). The T_{ntx} dataset (table) has four attributes, a_x, where $x = 1....4$.



Non-textual Dataset, $\mathbf{T_{ntx}}$

$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4}$
20	15	18	9
No	29	34	30
06	10	23	14
No	13	30	10
03	[illegible]	[illegible]	23
No	30	32	28
09	24	17	20

Non-textual Data, $\mathbf{t_{ntx}}$

$\mathbf{a_1}$: Rainfall
$\mathbf{a_2}$: Temperature at morning
$\mathbf{a_3}$: Temperature at afternoon
$\mathbf{a_4}$: Temperature at night

Textual Data, $\mathbf{t_{tx}}$
Temperature 13 sharply increase to 30 by afternoon and decrease to 10 by midnight with zero rainfall

Figure 1: Basic input-output structure of our system

3.2 Requirement analysis

Being a supervised statistical model, the system needs a parallel corpus for training purpose which should be a collection of non-textual tuple data and the corresponding textual data. We make an assumption here, that most of the attribute values present in any non-textual data will also appear in the corresponding textual data for that non-textual data. For our experiments we used the Prodigy-METEO (Belz, 2009) parallel corpus on wind-speed forecast data.

3.3 Design and development

To generate human readable and easily understandable textual data from non-textual data, an NLG system must ensure two criteria. Firstly, natural language text output should be related to the corresponding non-textual data's topic, i.e., output text must contain appropriate information. Secondly, the output text must be fluent, i.e., the output text must ensure its linguistic quality.

In the design step, we divide our system into two modules; one module holds the responsibility of ensuring informativeness and the other module maintains linguistic quality of the generated output text.

3.3.1 Informativeness Management Module

We define informative quality of a generated output text by considering how many attribute values of its corresponding non-textual data are present in the generated output text and in which order. It can be noted that if a given parallel corpus holds our requirement analysis criteria (cf. Section 3.2) then we can represent each textual data as a sequence of attribute-names (which should later be replaced with attribute-values) of its corresponding non-textual data with interlinked word-groups between two adjacent attribute values present in the sequence. This concept is illustrated in figure 2. Figure 2 shows the steps which are necessary for maintaining informativeness of the output textual data. The example shown in Figure 2 is taken from a non-textual tuple formed data from a temperature and rainfall weather dataset which is not our actual experimental dataset; it is presented only for illustration purpose.

We subdivide the informativeness module into two submodules. First submodule predicts the appropriate sequence of attribute names while the second submodule's job is to select all the interlinked word-groups that should be present in the sequence predicted by the first submodule.

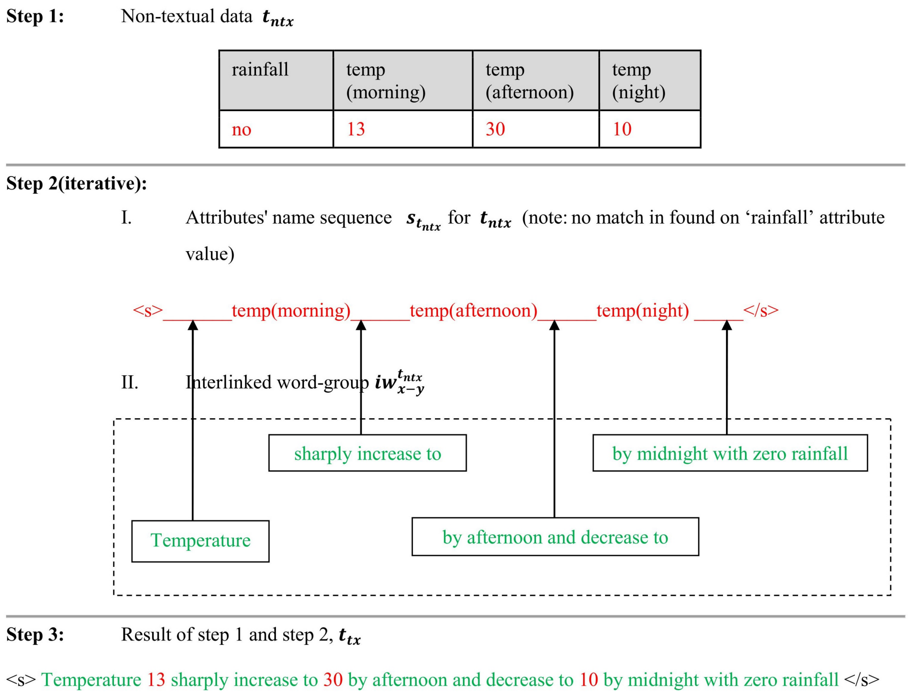

Step 1:
Non-textual data t_{ntx}
rainfall
temp (morning)
temp (afternoon)
temp (night)
no
13
30
10
Step 2(iterative):
I. Attributes' name sequence $s_{t_{ntx}}$ for t_{ntx} (note: no match in found on 'rainfall' attribute value)
<s>______temp(morning)_____temp(afternoon)_____temp(night) _____</s>
II. Interlinked word-group $iw^{t_{ntx}}_{x-y}$
sharply increase to
by midnight with zero rainfall
Temperature
by afternoon and decrease to
Step 3:
Result of step 1 and step 2, t_{tx}
<s> Temperature 13 sharply increase to 30 by afternoon and decrease to 10 by midnight with zero rainfall </s>

Figure 2: Steps for maintaining informativeness in the system

Let us consider generating text t_{tx} from a given non-textual data t_{ntx}. To achieve this we first need to find out the attributes' name sequence $s_{t_{ntx}}$ and thereafter identify all interlinked word-groups $iw^{t_{ntx}}_{*-*}$. Mathematically we can express this as in Equation 1 since t_{tx} is made up of $s_{t_{ntx}}$ and $iw^{t_{ntx}}_{*-*}$.

$$P(t_{tx}|t_{ntx}) = P(s_{t_{ntx}}, iw^{t_{ntx}}_{*-*}|t_{ntx}) \qquad (1)$$

$$P(t_{tx}|t_{ntx}) = P(s_{t_{ntx}}|t_{ntx}) * P(iw^{t_{ntx}}_{*-*}|t_{ntx}, s_{t_{ntx}}) \qquad (2)$$

Equation 2 rewrites Equation 1 as the product of two individual models where the first model, $P(s_{t_{ntx}}|t_{ntx})$, denotes prediction of attribute name sequence $s_{t_{ntx}}$ for the non-textual data t_{ntx}, while the second model, $P(iw^{t_{ntx}}_{*-*}|t_{ntx}, s_{t_{ntx}})$, denotes prediction of all the interlinked word-groups $iw^{t_{ntx}}_{*-*}$ for t_{ntx} in the attribute name sequence $s_{t_{ntx}}$ predicted by the first model.

i. **Predicting Attribute Name Sequence**
This model predicts the probable attribute name sequence for a given tuple-formed non-textual data. For this prediction, firstly each attribute value of a non-textual data needs to be identified in the corresponding textual data. It may so happen that some of the attribute values of the non-textual data might not be present in the corresponding textual data. However, we must add those attribute values (as features) in predicting attribute name sequence since they can play a crucial role in that prediction. After identification of all the attribute values present in each training textual data, we can train a model on this dataset which can take a new non-textual data and identify the corresponding textual data's attribute sequence. After predicting the attribute name sequence for a test non-textual data, we replace the attributes' names with their corresponding values.

Let us consider that we want to find the attribute name sequence $s_{t_{ntx}}$ corresponding to some non-textual data t_{ntx}. Let the non-textual dataset T_{ntx} ($t_{ntx} \in T_{ntx}$) contain $a_1, a_2, ...a_n$ attributes and the corresponding attribute values in t_{ntx} are $a_1^{t_{ntx}}, a_2^{t_{ntx}}, ...a_n^{t_{ntx}}$. Then we can express the attribute name sequence prediction for t_{ntx} as given in Equation 3.

$$P(s_{t_{ntx}}|t_{ntx}) =$$

$$P(s_{t_{ntx}}|a_1^{t_{ntx}}, a_2^{t_{ntx}}, ..., a_n^{t_{ntx}}) \quad (3)$$

ii. **Predicting interlinked word-groups**
For a sequence containing n possible attribute names/values corresponding to a textual data, there will be $n + 1$ number of interlinked word(s) (or word-groups) as we introduce two default pseudo-attributes, one at the start and the other at the end of the text.

We predict the interlinking word-group between two attribute names along the attribute name sequence predicted in the earlier step for a test textual data from a context window of six attribute names around the two attribute names currently being considered. Therefore, predicting the interlinked word-groups for a textual data are considered to be independent of each other. We also consider the attribute names of the non-textual data which do not appear in the attribute name sequence. Let us consider that we want to predict the interlinking word-groups $iw_{*-*}^{t_{ntx}}$ for an attribute name sequence $s_{t_{ntx}}$ of a non-textual data t_{ntx}. Let this $s_{t_{ntx}}$ sequence be $[...a_l\text{\_\_}a_m\text{\_\_}a_n\text{\_\_}a_o\text{\_\_}a_r\text{\_\_}a_s\text{\_\_}a_t\text{\_\_}a_u...]$ and we want to determine the intermediate word-group between a_o and a_r. Let us also assume that some of the attributes' (a_e, a_f, a_g) values of t_{ntx} are not present in $s_{t_{ntx}}$, which are $a_e^{t_{ntx}}$, $a_f^{t_{ntx}}$ and $a_g^{t_{ntx}}$. We model this task of predicting interlinking word-group between a_o and a_r for t_{ntx} and $s_{t_{ntx}}$ as in Equation 4.

$$P(iw_{o-r}^{t_{ntx}}|t_{ntx}, s_{t_{ntx}}) =$$

$$P(iw_{o-r}^{t_{ntx}}|a_m^{t_{ntx}}, a_n^{t_{ntx}}, a_o^{t_{ntx}}, a_r^{t_{ntx}}, a_s^{t_{ntx}}, a_t^{t_{ntx}}, a_e^{t_{ntx}}, a_f^{t_{ntx}}, a_g^{t_{ntx}}) \quad (4)$$

More precisely we can write the $P(iw_{*-*}^{t_{ntx}}|t_{ntx}, s_{t_{ntx}})$ term as the product of independent interlinking word-group prediction tasks as in Equation 5, where $prev$ (previous) and $next$ (next) are any of two adjacent attribute names in $s_{t_{ntx}}$.

$$P(iw_{*-*}^{t_{ntx}}|t_{ntx}, s_{t_{ntx}}) =$$

$$\prod P(iw_{(prev)-(next)}^{t_{ntx}}|t_{ntx}, s_{t_{ntx}}) \quad (5)$$

Therefore, Equation 2 can be rewritten as in Equation 6.

$$P(t_{tx}|t_{ntx}) = P(s_{t_{ntx}}, iw_{*-*}^{t_{ntx}}|t_{ntx})$$

$$= P(s_{t_{ntx}}|t_{ntx}) * P(iw_{*-*}^{t_{ntx}}|t_{ntx}, s_{t_{ntx}})$$

$$= P(s_{t_{ntx}}|t_{ntx}) * \prod P(iw_{(prev)-(next)}^{t_{ntx}}|t_{ntx}, s_{t_{ntx}}) \quad (6)$$

3.3.2 Linguistic Quality Management Module

The informativeness management module tries to answer "what will be content within the output textual data ", but it does not concern the linguistic quality of the generated text, e.g., fluency, readability, etc. In the linguistic quality management module we try to deal with the deficiency of linguistic quality in the generated textual data. Statistical language modelling is a well established technique for ensuring fluency and readability in natural language text. Therefore, as a final component, we incorporate a language model in our system. Hence, to maintain both informativeness and linguistic quality of the generated textual data, we model the task of NLG as given in Equation 7, where $PP(x)$ stands for the perplexity of string x.

$$P(t_{tx}|t_{ntx}) = P(s_{t_{ntx}}|t_{ntx})$$

$$* \prod P(iw_{(prev)-(next)}^{t_{ntx}}|t_{ntx}, s_{t_{ntx}}) * PP^{-1}(t_{tx}) \quad (7)$$

For our experiments, we trained a trigram language model on the training set textual data with a minor modification by replacing the attribute values with their attribute names.

3.3.3 Decoding

The search space of the NLG problem as modelled by Equation 7 is enormous. For example, if we consider top ten attribute name sequences, and for each attribute name sequence there are overall fifteen interlinked word-groups and for selecting each of these interlinked word-group we consider only top five candidates, then the search space for the generation task will contain $10 * 5^{15}$ candidates. To reduce the size of the search space and keep the computation problem tractable, we implemented the task of text generation as modelled in Equation 7 using *stack decoding* (Jelinek, 1969) with histogram pruning which limits the number of most promosing hypotheses to be explored by the size of the stack. In stack decoding with histogram pruning, the stack at any point of time contains only N (size of the stack) most promising partial hypotheses and during hypothesis expansion, a partial hypothesis is placed on the stack provided there is space in the stack, or, it is more promising than at least one of the partial hypotheses already stored in the stack. In case of stack overflow, the least promising hypothesis is discarded.

4 Experiments

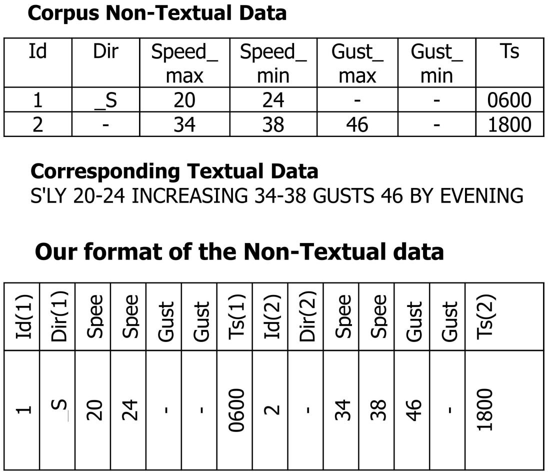
This section presents the dataset used in our experiments and the evaluation results of our system compared to some other NLG systems.

4.1 Dataset

As mentioned in Section 3.2, a non-textual–textual parallel dataset is required to train our system. The parallelism should be in such a form that each non-textual data can be represented as a tuple of attribute value instances and most of those attribute values should be present in its corresponding textual data.

We used the Prodigy-METEO[2] corpus (Belz, 2009), a wind forecast dataset, for our experiment. In the Prodigy-METEO corpus a single pair of non-textual–textual data stands for a particular day's wind forecast report. A non-textual data in that dataset is represented by a seven-component vector, where each component expresses a particular feature of wind data measurement at a moment of time. The seven components belong to a vector represented by [*id, direction, speed_min, speed_max, gust-speed_min, gust-speed_max, time*]. In that vector representation *id* stands for identification of the vector, *direction* mentions the wind speed direction, *speed_max and speed_min* denote the maximum and mnimum wind speed respectively, *gust_max and gust_min* represent the maximum and minimum wind gust speed respectively, and the last component *time* denotes the specific time instance when rest of components' readings were measured. For example, 1st April, 2001 wind forecast data is represented in this dataset as [[1,_SW,28,32,42,-,0600],[2,-,20,25,-,-,0000]] , where '-' represents a missing reading value.

As mentioned earlier our proposed model can process only a single tuple formed non-textual data at a time. However, the Prodigy-METEO corpus represents each non-textual data (wind forecast data for a particular day) by a sequence of multi-component vectors. For this reason, we merge all the vectors of a particular day's wind forecast data into a single tuple formed data. The merging of a particular day's wind forecast data vectors is illustrated in Figure 3. The Prodigy-METEO corpus comes with five predefined splits each of which has on an average 490 pairs of non-textual tuple data and the corresponding textual data.



Corpus Non-Textual Data

Id	Dir	Speed_ max	Speed_ min	Gust_ max	Gust_ min	Ts
1	_S	20	24	-	-	0600
2	-	34	38	46	-	1800

Corresponding Textual Data

S'LY 20-24 INCREASING 34-38 GUSTS 46 BY EVENING

Our format of the Non-Textual data

Id(1)	Dir(1)	Spee	Spee	Gust	Gust	Ts(1)	Id(2)	Dir(2)	Spee	Spee	Gust	Gust	Ts(2)
1	_S	20	24	-	-	0600	2	-	34	38	46	-	1800

Figure 3: Transformation of Prodigy-METEO corpus non-textual data representation for the system's input

4.2 Evaluation

We evaluated our system using both automatic evaluation metrics and human evaluation. For both human and automatic evaluation, we compared our system with ten existing NLG systems whose outputs on the Prodigy-METEO testset are also available in the Prodigy-METEO corpus. These ten NLG systems are PCFG-Greedy, PSCFG-Semantic, PSCFG-Unstructured, PCFG-Viterbii, PCFG-2gram, PCFG-Roulette, PBSMT-Unstructured,

[2] http://www.nltg.brighton.ac.uk/home/Anja.Belz/Prodigy

Non-Textual Data	[[1,_SW-WSW,08,12,-,-,0600],[2,_VAR,02,06,-,-,1800]]
Corpus	SW-WSW 08-12 FALLING VARIABLE 02-06 BY LATE AFTERNOON
SumTime	SW-WSW 10 OR LESS BECOMING VARIABLE BY LATE AFTERNOON
PSCFG-Semantic	SW-WSW 08-12 FALLING VARIABLE 02-06 BY EVENING
PSCFG-Unstructured	SW-WSW 08-12 FALLING VARIABLE 02-06 BY EVENING
Our Proposed System	**SW-WSW 08-12 FALLING VARIABLE 02-06 BY EVENING**
PCFG-Greedy	SW-WSW 8-12 THEN FALLING VARIABLE 2-6 BY EVENING
PCFG-Viterbi	SW-WSW 8-12 THEN FALLING VARIABLE 2-6
PCFG-Roulette	SW-WSW 8-12 FALLING VARIABLE 2-6
PCFG-2gram	SW-WSW 8-12 THEN VARIABLE 2-6
PCFG-Random	SOON SW-WSW 8-12 THEN STEADILY EASING TO VARIABLE'LY 2-6 AHEAD OF THE FRONT FOR A TIME THIS AFTERNOON
PBSMT-Unstructured	LESS SW-WSW 08-12 GRADUALLY FALLING VARIABLE 02-06 BY EVENING
PBSMT-Structured	GUSTS SW-WSW 08-12 BY IN TO AND FALLING TO UNKNOWN VARIABLE 02-06 BY EVENING

Figure 4: An sample of input and outputs of different NLG system

SumTime-Hybrid, PBSMT-Structured and PCFG-Random (Belz and Kow, 2009). Figure 4 shows a sample input and outputs of all the above mentioned systems including our system.

4.2.1 Automatic Evaluation

For automatic evaluation, we used two automatic evaluation metrics; BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005). Both BLEU and METEOR were originally proposed for evaluation of machine translation (MT) systems However, due to the similarity between the two tasks (i.e., MT and NLG) from the point of view of their working principles, most of the NLG systems are also evaluated using these two automatic MT evaluation metrics.

Because of the relatively small size of the dataset, we took a five-fold cross validation policy which was predefined in the Prodigy-METEO corpus. Table 1 presents the evaluation results obtained with BLEU and METEOR on our system along with the ten other NLG systems.

System	BLEU score	Meteor score
Corpus	1	1
PCFG-Greedy	0.65	0.85
PSCFG-Semantic	0.64	0.83
PSCFG-Unstructured	0.62	0.81
Proposed System	**0.61**	**0.82**
PCFG-Viterbii	0.57	0.76
PCFG-2gram	0.56	0.76
PCFG-Roulette	0.52	0.76
PBSMT-Unstructured	0.51	0.81
SumTime-Hybrid	0.46	0.67
PBSMT-Structure	0.34	0.59
PCFG-Random	0.28	0.52

Table 1: Comparison using automatic metric evaluation

4.2.2 Human-based Evaluation

Evaluation using automatic evaluation metrics is very popular among researchers and developers since automatic evaluation is very fast and cheap. Automatic evaluation metrics are good indicators of system performance and they greatly help day-to-day system development. However, despite being very time intensive and costly, human evaluation still serves as the de-facto evaluation standard and the worth of automatic evaluation metrics are typically judged based on how well they correlate with human evaluation.

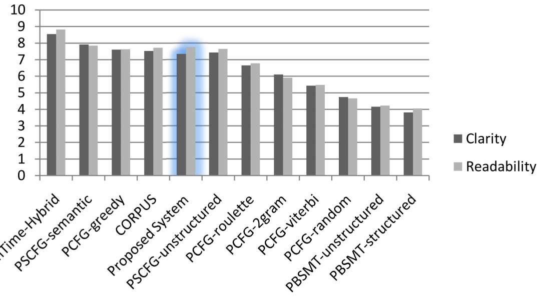


Figure 5: Comparison using human evaluation

We also evaluated the systems using human evaluation on a part of the test dataset. We carried out human evaluation to measure the clarity and readability of the texts generated by the NLG systems. Clarity measures truthfulness and correctness of a textual data whereas readability concerns fluency of the textual data. 30 instances (out of total 232) were randomly chosen from the testset for the pilot human evaluation and the output from 11 different systems along with the corresponding non-textual data were presented to the human evaluators. Five students from different backgrounds who acted as human evaluators were asked to rate 72 outputs each in a 10 point scale. The output of human evaluation is presented in Figure 5.

4.3 Result Analysis

As per the outcomes of automatic evaluation, our system provided the third and fourth best results in METEOR and BLEU, respectively. According to METEOR, our system is only behind the PCGF-Greedy and PSCFG-Semantic systems while according to BLEU, PCFG-greedy, PSCFG-Semantic and PSCFG-Unstructured systems perform better than our proposed model. However, these systems which are ahead of our system in performance as per automatic evaluation are not fully automatic, whereas our system does not require any human effort or additional knowledge other than a non-textual–textual parallel corpus.

Human evaluation preferred rule based systems over automatic knowledge-light systems. The SumTime system, which is a rule based system and is placed in the ninth position according to both BLEU and METEOR, is adjudged the best system in human evaluation. Our system ranks fourth among the 11 systems according to human evaluation. The gold standard reference set was also provided to the human evaluators for evaluation without their knowledge and, surprisingly, the reference set was ranked fourth.

We calculated Pearson correlation coefficient between scores produced by automatic evaluation metrics (BLEU and METEOR) and human evaluation. For human evaluation we considered the average of clarity and readability. The correlation coefficients are r(Human, Bleu)=0.61 and r(Human, METEOR)=0.57 .

5 Conclusions

The statistical NLG system presented in this paper does not require any external agents' involvement. It is a domain independent system and can easily be shifted from one application domain to another without any change.

To avail good quality output text from the system, one must conform to the requirement specified in Section 3.2. The NLG system will perform accurately if all attributes present in the training tuple-formed non-textual data contain distinct values. If this criterion can be assured, then it will be trivial to match each of those attribute values present in the non-textual data to their appearance in the corresponding textual data. However, if this constraint is not possible to be satisfied the system will still work. It is worth to be mentioned here that in cases when not a single attribute value of non-textual data can be found in the corresponding textual data, then our system will behave like an instance based NLG system.

Acknowledgments

This work is partially supported by Media Lab Asia, Department of Electronics and Information Technology (DeitY), Government of India, under the Young Faculty Research Fellowship of the Visvesvaraya PhD Scheme for Electronics & IT and through the project "CLIA System Phase II".

References

Ibrahim Adeyanju. 2012. Article: Generating weather forecast texts with case based reasoning. *International Journal of Computer Applications*, 45(10):35–40, May. Full text available.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. pages 65–72.

Anja Belz and Eric Kow. 2009. System building cost vs. output quality in data-to-text generation. In *Proceedings of the 12th European Workshop on Natural Language Generation*, ENLG '09, pages 16–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Nat. Lang. Eng.*, 14(4):431–455, October.

Anja Belz. 2009. Prodigy-meteo: Pre-alpha release notes (nov 2009).

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311, June.

Eli Goldberg, Norbert Driedger, and Richard I. Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert: Intelligent Systems and Their Applications*, 9(2):45–53, April.

Nizar Habash, 2000. *Envisioning Machine Translation in the Information Future: 4th Conference of the Association for Machine Translation in the Americas, AMTA 2000 Cuernavaca, Mexico, October 10–14, 2000 Proceedings*, chapter Oxygen: A Language Independent Linearization Engine, pages 68–79. Springer Berlin Heidelberg, Berlin, Heidelberg.

F. Jelinek. 1969. Fast sequential decoding algorithm using a stack. *IBM J. Res. Dev.*, 13(6):675–685, November.

Kevin Knight and Vasileios Hatzivassiloglou. 1995. Two-level, many-paths generation. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, pages 252–260, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

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*, pages 704–710, Montreal, Quebec, Canada, August. Association for Computational Linguistics.

Brian Langner and Alan W Black. 2009. Mountain: a translation-based approach to natural language generation for dialog systems.

Kathleen McKeown, Karen Kukich, and James Shaw. 1994. Practical issues in automatic documentation generation. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 7–14, Stuttgart, Germany, October. Association for Computational Linguistics.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Alice H. Oh and Alexander I. Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational Systems - Volume 3*, ANLP/NAACL-ConvSyst '00, pages 27–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shimei Pan and James Shaw. 2004. Segue: A hybrid case-based surface natural language generator. In *In Proceedings of INLG 2004*, pages 130–140.

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. 1997. Building applied natural language generation systems. *Nat. Lang. Eng.*, 3(1):57–87, March.

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artif. Intell.*, 167(1-2):137–169, September.

Ilya Sutskever, James Martens, and Geoffrey E. Hinton. 2011. Generating text with recurrent neural networks. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, New York, NY, USA. ACM.

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-hao Su, David Vandyke, and Steve J. Young. 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *CoRR*, abs/1508.01755.