A Statistical NLG Framework for Aggregated Planning and Realization

Ravi Kondadadi*, Blake Howald and Frank Schilder
Thomson Reuters, Research & Development
610 Opperman Drive, Eagan, MN 55123
firstname.lastname@thomsonreuters.com

Abstract

We present a hybrid natural language generation (NLG) system that consolidates macro and micro planning and surface realization tasks into one statistical learning process. Our novel approach is based on deriving a template bank automatically from a corpus of texts from a target domain. First, we identify domain specific entity tags and Discourse Representation Structures on a per sentence basis. Each sentence is then organized into semantically similar groups (representing a domain specific concept) by *k*-means clustering. After this semi-automatic processing (human review of cluster assignments), a number of corpus–level statistics are compiled and used as features by a ranking SVM to develop model weights from a training corpus. At generation time, a set of input data, the collection of semantically organized templates, and the model weights are used to select optimal templates. Our system is evaluated with automatic, non–expert crowdsourced and expert evaluation metrics. We also introduce a novel automatic metric – *syntactic variability* – that represents linguistic variation as a measure of unique template sequences across a collection of automatically generated documents. The metrics for generated *weather* and *biography* texts fall within acceptable ranges. In sum, we argue that our statistical approach to NLG reduces the need for complicated knowledge-based architectures and readily adapts to different domains with reduced development time.

*Ravi Kondadadi is now affiliated with Nuance Communications, Inc.

1 Introduction

NLG is the process of generating natural-sounding text from non-linguistic inputs. A typical NLG system contains three main components: (1) Document (Macro) Planning - deciding what content should be realized in the output and how it should be structured; (2) Sentence (Micro) planning - generating a detailed sentence specification and selecting appropriate referring expressions; and (3) Surface Realization - generating the final text after applying morphological modifications based on syntactic rules (*see e.g.*, Bateman and Zock (2003), Reiter and Dale (2000) and McKeown (1985)). However, document planning is arguably one of the most crucial components of an NLG system and is responsible for making the texts express the desired communicative goal in a coherent structure. If the document planning stage fails, the communicative goal of the generated text will not be met even if the other two stages are perfect. While most traditional systems simplify development by using a pipelined approach where (1-3) are executed in a sequence, this can result in errors at one stage propagating to successive stages (*see e.g.,* Robin and McKeown (1996)). We propose a hybrid framework that combines (1-3) by converting data to text in one single process.

Most NLG systems fall into two broad categories: knowledge-based and statistical. Knowledge-based systems heavily depend on having domain expertise to come up with handcrafted rules at each stage of a pipeline. Although knowledge-based systems can produce high quality text, they are (1) very expensive to build, involving a lot of discussion with the end users of the system for the document planning stage alone; (2) have limited linguistic coverage, as it is time consuming to capture linguistic variation; and (3) one has to start from scratch for each new domain because the developed components cannot be reused.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1406–1415,
Sofia, Bulgaria, August 4-9 2013.

Statistical systems, on the other hand, are fairly inexpensive, more adaptable and rely on having historical data for the given domain. Coverage is likely to be high if more historical data is available. The main disadvantage with statistical systems is that they are more prone to errors and the output text may not be coherent as there are less constraints on the generated text.

Our framework is a hybrid of statistical and template-based systems. Many knowledge-based systems use templates to generate text. A template structure contains "gaps" that are filled to generate the output. The idea is to create a lot of templates from the historical data and select the right template based on some constraints. To the best of our knowledge, this is the first hybrid statistical-template-based system that combines all three stages of NLG. Experiments with different variants of our system (for *biography* and *weather* subject matter domains) demonstrate that our system generates reasonable texts.

Also, in addition to the standard metrics used to evaluate NLG systems (e.g., BLEU, NIST, etc.), we present a unique text evaluation metric called *syntactic variability* to measure the linguistic variation of generated texts. This metric applies to the document collection level and is based on computing the number of unique template sequences among all the generated texts. A higher number indicates the texts are more variable and natural-sounding whereas a lower number shows they are more redundant. We argue that this metric is useful for evaluating template-based systems and for *any* type of text generation for domains where linguistic variability is favored (e.g., the user is expected to go through more than one document in the same session).

The main contributions of this paper are (1) A statistical NLG system that combines document and sentence planning and surface realization into one single process; and (2) A new metric – *syntactic variability* – is proposed to measure the syntactic and morphological variability of the generated texts. We believe this is the first work to propose an automatic metric to measure linguistic variability of generated texts in NLG.

Section 2 provides an overview of related work on NLG. We present our main system in Section 3. The system is evaluated and discussed in Section 4. Finally, we conclude in Section 5 and point out future directions of research.

2 Background

Typically, knowledge-based NLG systems are implemented by rules and, as mentioned above, have a pipelined architecture for the document and sentence planning stages and surface realization (Hovy, 1993; Moore and Paris, 1993). However, document planning is arguably the most important task (Sripada et al., 2001). It follows that approaches to document planning are rule-based as well and, concomitantly, are usually domain specific. For example, Bouayad-Agha, et al. (2011) proposed document planning based on an ontology knowledge base to generate football summaries. For rule–based systems, rules exist for selecting content to grammatical choices to post-processing (e.g., pronoun generation). These rules are often tailored to a given system, with input from multiple experts; consequently, there is a high associated development cost (e.g., 12 person months for the SUMTIME-METEO system (Belz, 2007)).

Statistical approaches can reduce extensive development time by relying on corpus data to "learn" rules for one or more components of an NLG system (Langkilde and Knight, 1998). For example, Duboue and McKeown (2003) proposed a statistical approach to extract content selection rules for biography descriptions. Further, statistical approaches should be more adaptable to different domains than their rule-based equivalents (Angeli et al., 2012). For example, Barzilay and Lapata (2005) formulated content selection as a classification task to produce football summaries and Kelly et al. (2009) extended Barzilay and Lapata's approach for generating match reports for cricket.

The present work builds on Howald et al. (2013) where, in a given corpus, a combination of domain specific named entity tagging and clustering sentences (based on semantic predicates) were used to generate templates. However, while the system consolidated both sentence planning and surface realization with this approach (described in more detail in Section 3), the document plan was given via the input data and sequencing information was present in training documents. For the present research, we introduce a similar method that leverages the distributions of document–level features in the training corpus to incorporate a statistical document planning component. Consequently, we are able to create a streamlined statistical NLG architecture that balances natural

human–like variability with appropriate and accurate information.

3 Methodology

In order to generate text for a given domain our system runs input data through a statistical ranking model to select a sequence of templates that best fit the input data (E). In order to build the ranking model, our system takes historical data (corpus) for the domain through four components: (A) preprocessing; (B) "conceptual unit" creation; (C) collecting statistics; and (D) ranking model building (summarized in Figure 1). In this section, we describe each component in detail.

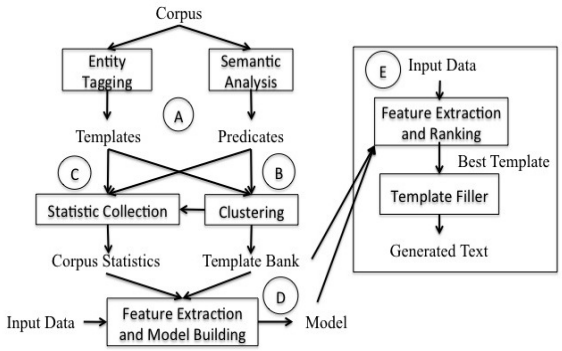

Figure 1: System Architecture.

3.1 Preprocessing

The first component processes the given corpus to extract templates. We assume that each document in the corpus is classified to a specific domain. Preprocessing involves uncovering the underlying semantic structure of the corpus and using this as a foundation for template creation (Lu et al., 2009; Lu and Ng, 2011; Konstas and Lapata, 2012).

We first split each document in the corpus into sentences and create a shallow Discourse Representation Structure (following Discourse Representation Theory (Kamp and Reyle, 1993)) of each sentence. The DRS consists of semantic predicates and named entity tags. We use *Boxer* semantic analyzer (Bos, 2008) to extract semantic predicates such as EVENT or DATE. In parallel, domain specific named entity tags are identified and, in conjunction with the semantic predicates, are used to create templates. We developed the named-entity tagger for the *weather* domain ourselves. To tag entities in the *biography* domain, we used OpenCalais (www.opencalais.com). For example, in the biography in (1), the conceptual meaning (semantic predicates and domain-specific entities) of sentences (a-b) are represented in (c-d). The corresponding templates are showing in (e-f).

(1) *Sentence*
a. Mr. Mitsutaka Kambe has been serving as Managing Director of the 77 Bank, Ltd. since June 27, 2008.
b. He holds a Bachelor's in finance from USC and a MBA from UCLA.

Conceptual Meaning
c. SERVING | TITLE | PERSON | COMPANY | DATE
d. HOLDS | DEGREE | SUBJECT | INSTITUTION| EVENT

Templates
e. **[person]** has been serving as **[title]** of the **[company]** since **[date]**.
f. **[person]** holds a **[degree]** in **[subject]** from **[institution]** and a **[degree]** from **[institution]**.

The outputs of the preprocessing stage are the template bank and predicate information for each template in the corpus.[1]

3.2 Creating Conceptual Units

The next step is to create conceptual units for the corpus by clustering templates. This is a semi-automatic process where we use the predicate information for each template to compute similarity between templates. We use *k*-means clustering with *k* (equivalent to the number of semantic concepts in the domain) set to an arbitrarily high value (100) to over-generate (using the WEKA toolkit (Witten and Frank, 2005)). This allows for easier manual verification of the generated clusters and we merge them if necessary. We assign a unique identifier called a CuId (Conceptual Unit Identifier) to each cluster, which represents a "conceptual unit". We associate each template in the corpus to a corresponding CuId. For example, in (2), using the templates in (1e-f), the identified named entities are assigned to a clustered CuId (2a-b).

(2) *Conceptual Units*
a. $\{CuId : 000\}$ – **[person]** has been serving as **[title]** of the **[company]** since **[date]**.
b. $\{CuId : 001\}$ – **[person]** holds a **[degree]** in **[subject]** from **[institution]** and a **[degree]** from **[institution]**.

At this stage, we will have a set of conceptual units with corresponding template collections (see Howald et al. (2013) for a further explanation of Sections 3.1-3.2).

[1] A similar approach to the clustering of semantic content is found in Duboue and McKeown (2003), where text with stopwords removed were used as semantic input. Boxer provides a similar representation with the addition of domain general tags. However, to contrast our work from Duboue and McKeown, which focused on content selection, we are focused on learning templates from the semantic representations for the complete generation system (covering content selection, aggregation, sentence and document planning).

3.3 Collecting Corpus Statistics

After identifying the different conceptual units and the template bank, we collect a number of statistics from the corpus:

- Frequency distribution of templates overall and per position
- Frequency distribution of CuIds overall and per position
- Average number of entity tags by CuId as well as the entity distribution by CuId
- Average number of entity tags by position as well as the entity distribution by position
- Average number of words per CuId.
- Average number of words per CuId and position combination.
- Average number of words per position
- Frequency distribution of the main verbs by position
- Frequency distribution of CuId sequences (bigrams and trigrams only) overall and per position
- Frequency distribution of template sequences (bigrams and trigrams only) overall and per position
- Frequency distribution of entity tag sequences overall and per position
- The average, minimum, maximum number of CuIds across all documents

As discussed in the next section, these statistics are turned into features used for building a ranking model in the next component.

3.4 Building a ranking model

The core component of our system is a statistical model that ranks a set of templates for a given position (sentence 1, sentence 2, ..., sentence n) based on the input data. The input data in our tasks was extracted from a training document; this serves as a temporary surrogate to a database. The task is to learn the ranks of all the templates from all CuIds at each position.

To generate the training data, we first filter the templates that have named entity tags not specified in the input data. This will make sure the generated text does not have any unfilled entity tags. We then rank templates according to the Levenshtein edit distance (Levenshtein, 1966) from the template corresponding to the current sentence in the training document (using the top 10 ranked templates in training for ease of processing effort). We experimented with other ranking schemes such as entity-based similarity (similarity between entity sequences in the templates) and a combination of edit-distance based and entity-based similarities. We obtained better results with edit distance. For each template, we generate the following features to build the ranking model. Most of the features are based on the corpus statistics mentioned above.

- **CuId given position**: This is a binary feature where the current CuId is either the same as the most frequent CuId for the position (1) or not (0).
- **Overlap of named entities**: Number of common entities between current CuId and most likely CuId for the position
- **Prior template**: Probability of the sequence of templates selected at the previous position and the current template (iterated for the last three positions).
- **Prior CuId**: Probability of the sequence of the CuId selected at the previous position and the current CuId (iterated for the last three positions).
- **Difference in number of words**: Absolute difference between number of words for current template and average number of words for the CuId
- **Difference in number of words given position**: Absolute difference between number of words for current template and average number of words for CuId at given position
- **Percentage of unused data**: This feature represents the portion of the unused input so far.
- **Difference in number of named entities**: Absolute difference between the number of named entities in the current template and the average number of named entities for the current position
- **Most frequent verb for the position**: Binary valued feature where the main verb of the template belongs to the most frequent verb group given the position is either the same (1) or not (0).
- **Average number of words used**: Ratio of number of words in the generated text so far to the average number of words.
- **Average number of entities**: Ratio of number of named entities in the generated text so far to the average number of named entities.
- **Most likely CuId given position and previous CuId**: Binary feature indicating if the current CuId is most likely given the position and the previous CuId.
- **Similarity between the most likely template in CuId and current template**: Edit distance between the current template and the most likely template for the current CuId.
- **Similarity between the most likely template in CuId given position and current template**: Edit distance between the current template and the most likely template for the current CuId at the current position.

We used a linear kernel for a ranking SVM (Joachims, 2002) (*cost* set to total queries) to learn the weights associated with each feature for the different domains.

3.5 Generation

At generation time, our system has a set of input data, a semantically organized template bank (collection of templates organized by CuId) and a model from training on the documents for a given domain. We first filter out those templates that contain a named entity tag not present in the input data. Then, we compute a score for each of the remaining templates from the feature values and the feature weights from the model. The template with the highest overall score is selected and filled with matching entity tags from the input data and

appended to the generated text.

Before generating the next sentence, we track those entities used in the initial sentence generation and decide to either remove those entities from the input data or keep the entity for one or more additional sentence generations. For example, in the *biography* discourses, the name of the person may occur only once in the input data, but it may be useful for creating good texts to have that person's name available for subsequent generations. To illustrate in (3), if we remove *James Smithton* from the input data after the initial generation, an irrelevant sentence (d) is generated as the input data will only have one company after the removal of *James Smithton* and the model will only select a template with one company. If we keep *James Smithton*, then the generations in (a-b) are more cohesive.

(3) *Use more than once*

a. Mr. James Smithton was appointed CFO at Fordway Internation in April.

b. Previously, Mr. Smithton was CFO of the Keyes Development Group.

Use once and remove

c. Mr. James Smithton was appointed CFO at Fordway Internation in April.

d. Keyes Development Group is a venture capital firm.

Deciding on what type of entities and how to remove them is different for each domain. For example, some entities are very unique to a text and should not be made available for subsequent generations as doing so would lead to unwanted redundancies (*e.g.*, mentioning the name of current company in a biography discourse more than once as in (3)) and some entities are general and should be retained. Our system possesses the ability to monitor the data usage from historical data and we can set parameters (based on the distribution of entities) on the usage to ensure coherent generations for a given domain.

Once the input data has been modified (i.e., an entity have been removed, replaced or retained), it serves as the new input data for the next sentence generation. This process repeats until reaching the minimum number of sentences for the domain (determined from the training corpus statistic) and then continues until all of the remaining input data is consumed (and not to exceed the predetermined maximum number of sentences, also determined from the training corpus statistic).

4 Evaluation and Discussion

In this section, we first discuss the corpus data used to train and generate texts. Then, the results of both automatic and human evaluations of our system's generations against the original and baseline texts are considered as a means of determining performance. For all experiments reported in this section, the baseline system selects the most frequent conceptual unit at the given position, chooses the most likely template for the conceptual unit, and fills the template with input data. The above process is repeated until the number of sentences is less than or equal to the average number of sentences for the given domain.

4.1 Data

We ran our system on two different domains: corporate officer and director *biographies* and offshore oil rig *weather* reports from the SUMTIME-METEO corpus ((Reiter et al., 2005)). The *biography* domain includes 1150 texts ranging from 3-17 sentences and the *weather* domain includes 1045 weather reports ranging from 1-6 sentences.[2] We used a training-test(generation) split of 70/30.

(4) provides generation comparisons for the system (*_DocSys*), baseline (*_DocBase*) and original (*_DocOrig*) randomly selected text snippets from each domain. The variability of the generated texts ranges from a close similarity to slightly shorter - not an uncommon (Belz and Reiter, 2006), but not necessarily detrimental, observation for NLG systems (van Deemter et al., 2005).

(4) *Weather_DocOrig*

a. Another weak cold front will move ne to Cornwall by later Friday.

Weather_DocSys

b. Another weak cold front will move ne to Cornwall during Friday.

Weather_DocBase

c. Another weak cold front from ne through the Cornwall will remain slow moving.

Bio_DocOrig

d. He previously served as Director of Sales Planning and Manager of Loan Center.

Bio_DocSys

e. He previously served as Director of Sales in Loan Center of the Company.

Bio_DocBase

[2]The SUMTIME-METEO project is a common bench mark in NLG. However, we provide no comparison between our system and SUMTIME-METEO as our system utilized the generated forecasts from SUMTIME-METEO's system as the historical data. We cannot compare with other statistical generation systems like (Belz, 2007) as they only focussed on the part of the forecasts the predicts wind characteristics whereas our system generates the complete forecasts.

f. He previously served as Director of Sales of the Company.

The *DocSys* and *DocBase* generations are largely grammatical and coherent on the surface with some variance, but there are graded semantic variations (e.g., *Director of Sales Planning* vs. *Director of Sales* (4g-h) and *move ne to Cornwall* vs. *from ne through the Cornwall*). Both automatic and human evaluations are required in NLG to determine the impact of these variances on the understandability of the texts in general (non-experts) and as they are representative of particular subject matter domains (experts). The following sections discuss the evaluation results.

4.2 Automatic Metrics

We used BLEU–4 (Papineni et al., 2002), METEOR (v.1.3) (Denkowski and Lavie, 2011) to evaluate the texts at document level. Both BLEU–4 and METEOR originate from machine translation research. BLEU–4 measures the degree of 4-gram overlap between documents. METEOR uses a unigram weighted *f*–score less a penalty based on chunking dissimilarity. These metrics only evaluate the text on a document level but fail to identify "syntactic repetitiveness" across documents in a document collection. This is an important characteristic of a document collection to avoid banality. To address this issue, we propose a new automatic metric called *syntactic variability*. In order to compute this metric, each document should be represented as a sequence of templates by associating each sentence in the document with a template in the template bank. *Syntactic variability* is defined as the percentage of unique template sequences across all generated documents. It ranges between 0 and 1. A higher value indicates that more documents in the collection are linguistically different from each other and a value closer to zero shows that most of documents have the similar language despite different input data.[3]

As indicated in Figure 2, the BLEU-4 scores are low for all *DocSys* and *DocBase* generations (as compared to *DocOrig*) for each domain. However, the METEOR scores, while low overall (ranging from .201-.437) are noticeably increased over BLEU-4 (which ranges from .199-.320).

Given the nature of each metric, the results indicate that the generated and baseline texts have very different surface realizations compared to the originals (low BLEU-4), but are still capturing the content of the originals (higher METEOR). Both BLEU–4 and METEOR measure the similarity of the generated text to the original text, but fail to penalize repetitiveness across texts, which is addressed by the *syntactic variability* metric. There is no statistically significant difference between *DocSys* and *DocBase* generations for METEOR and BLEU–4.[4] However, there is a statistically significant difference in the *syntactic variability* metric for both domains (*weather* - χ^2=137.16, d.f.=1, $p<.0001$; *biography* - χ^2=96.641, d.f.=1, $p<.0001$) - the variability of the *DocSys* generations is greater than the *DocBase* generations, which shows that texts generated by our system are more variable than the baseline texts.

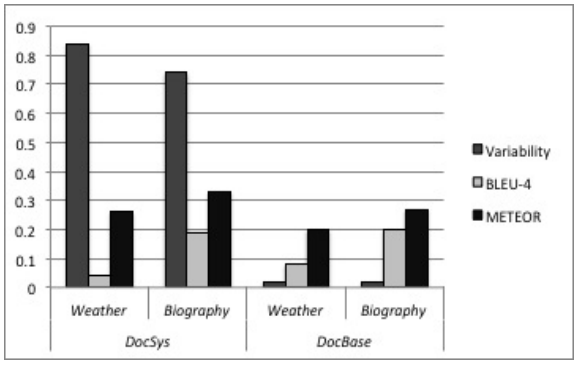

Figure 2: Automatic Evaluations.

The use of automatic metrics is a common evaluation method in NLG, but they must be reconciled against non–expert and expert level evaluations.

4.3 Non-Expert Human Evaluations

Two sets of crowdsourced human evaluation tasks (run on CrowdFlower) were constructed to compare against the automatic metrics: (1) an understandability evaluation of the entire text on a three-point scale: **Fluent** = no grammatical or informative barriers; **Understandable** = some grammatical or informative barriers; **Disfluent** = significant grammatical or informative barriers; and (2) a sentence–level preference between sentence pairs (e.g., "Do you prefer Sentence A (from *DocOrig*) or the corresponding Sentence B (from *DocBase/DocSys*)").

[3] Of course, syntactic and semantic repetitiveness could be captured by *syntactic variability*, but only if this is the nature of the underlying historical data - *financial* texts tend to be fairly repetitive.

[4] BLEU–4: *weather* - χ^2=1.418, d.f.=1, p=.230; *biography* - χ^2=0.311, d.f.=1, p=.354. METEOR: *weather* - χ^2=1.016, d.f.=1, p=.313; *biography* - χ^2=0.851, d.f.=1, p=.354.

Over 100 native English speakers contributed, each one restricted to providing no more than 50 responses and only after they successfully answered 4 "gold data" questions correctly. We also omitted those evaluators with a disproportionately high response rate. No other data was collected on the contributors (although geographic data (country, region, city) and IP addresses were available). For the sentence–level preference task, the pair orderings were randomized to prevent click bias.

For the text–understandability task, 40 documents were chosen at random from the *DocOrig* test set along with the corresponding 40 *DocSys* and *DocBase* generations (240 documents total/120 for each domain). 8 judgments per document were solicited from the crowd (1920 total judgments, 69.51 average agreement) and are summarized in Figures 3 and 4 (*biography* and *weather* respectively).

If the system is performing well and the ranking model is actually contributing to increased performance, the accepted trend should be that the *DocOrig* texts are more fluent and preferred compared to both the *DocSys* and *DocBase* systems. However, the differences between *DocOrig* and *DocSys* will not be significant, the differences between *DocOrig* and *DocBase* and *DocSys* and *DocBase* **will** be significantly different.

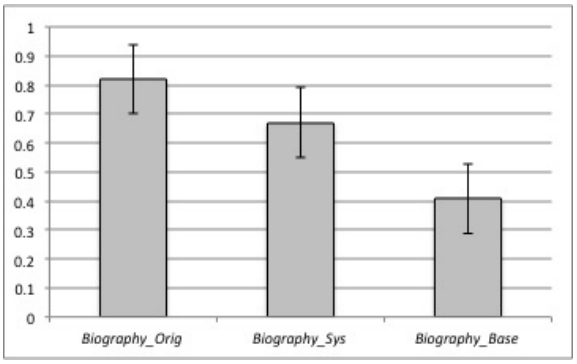

Figure 3: Biography Text Evaluations.

Focusing on fluency ratings, it is expected that the *DocOrig* generations will have the highest fluency (as they are human generated). Further, if the *DocSys* is performing well, it is expected that the fluency rating will be less than the *DocOrig* and higher than *DocBase*. Figure 3, which shows the *biography* text evaluations, demonstrates this acceptable distribution of performances.

For the *weather* discourses, as evident from Figure 4, the acceptable trend holds between the *DocSys* and *DocBase* generations, and the *DocSys* generation fluency is actually slightly higher than *DocOrig*. This is possibly because the *DocOrig* texts are from a particular subject matter - weather forecasts for offshore oil rigs in the U.K. - which may be difficult for people in general to understand. Nonetheless, the demonstrated trend is favorable to our system.

In terms of significance, there are no statistically significant differences between the systems for *weather* (*DocOrig vs. DocSys* - χ^2=.347, d.f.=1, p=.555; *DocOrig vs. DocBase* - χ^2=.090, d.f.=1, p=.764; *DocSys vs. DocBase* - χ^2=.790, d.f.=1, p=.373). While this is a good result for comparing *DocOrig* and *DocSys* generations, it is not for the other pairs. However, numerically, the trend is in the right direction despite not being able to demonstrate significance. For *biography*, the trend fits nicely both numerically and in terms of statistical significance (*DocOrig vs. DocSys* - χ^2=5.094, d.f.=1, p=.024; *DocOrig vs. DocBase* - χ^2=35.171, d.f.=1, $p<.0001$; *DocSys vs. DocBase* - χ^2=14.000, d.f.=1, $p<.0001$).

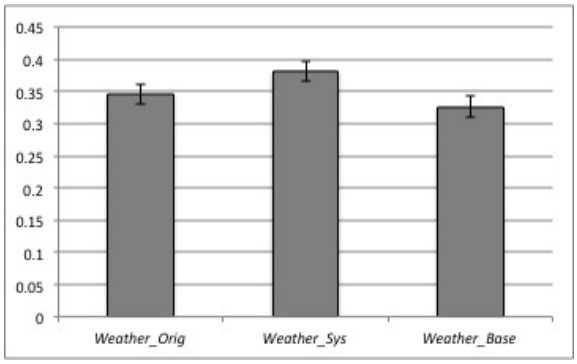

Figure 4: Weather Text Evaluations.

For the sentence preference task, equivalent sentences across the 120 documents were chosen at random (80 sentences from *biography* and 74 sentences from *weather*). 8 judgments per comparison were solicited from the crowd (3758 total judgments, 75.87 average agreement) and are summarized in Figures 5 and 6 (*biography* and *weather*, respectively).

Similar to the text–understandability task, an acceptable performance pattern should include the *DocOrig* texts being preferred to both *DocSys* and *DocBase* generations and the *DocSys* generation preferred to the *DocBase*. The closer the *DocSys* generation is to the *DocOrig*, the better *DocSys* is performing. The *biography* domain illus-

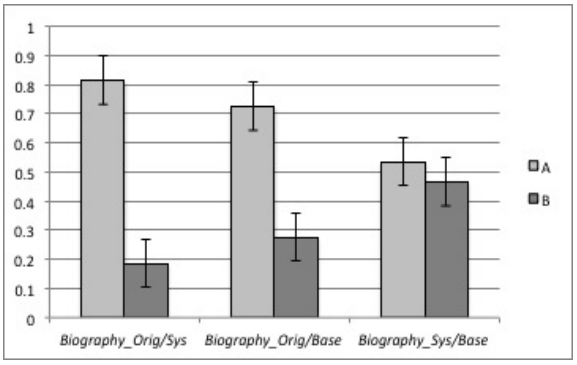

Figure 5: Biography Sentence Evaluations.

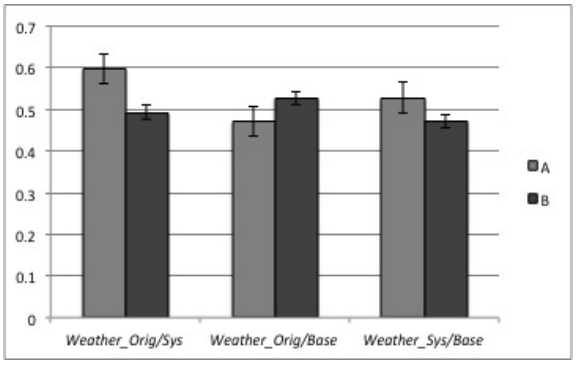

Figure 6: Weather Sentence Evaluations.

trates this scenario (Figure 5) where the results are similar to the text-understandability experiments. In contrast, for *weather* domain, sentences from *DocBase* system were preferred to our system's (Figure 6). We looked at the cases where the preferences were in favor of *DocBase*. It appears that because of high *syntactic variability*, our system can produce quite complex sentences where as the non-experts seem to prefer shorter and simpler sentences because of the complexity of the text.

In terms of significance, there are no statistically significant differences between the systems for *weather* (*DocOrig vs. DocSys* - χ^2=6.48, d.f.=1, p=.011; *DocOrig vs. DocBase* - χ^2=.720, d.f.=1, p=.396; *DocSys vs. DocBase* - χ^2=.720, d.f.=1, p=.396). The trend is different compared to the fluency metric above in that the *DocBase* system is outperforming the *DocOrig* generations to an almost statistically significant difference - the remaining comparisons follow the trend. We believe that this is for similar reasons stated above - i.e., the generation may be a more digestible version of a technical document. More problematic is the results of the *biography* evaluations. Here there is a statistically significant difference between the *DocSys* and *DocOrig* and no statistically significant difference between the *DocSys* and *DocBase* generations (*DocOrig vs. DocSys* - χ^2=76.880, d.f.=1, p<.0001; *DocOrig vs. DocBase* - χ^2=38.720, d.f.=1, p<.0001; *DocSys vs. DocBase* - χ^2=.720, d.f.=1, p=.396). Again, this distribution of preferences is numerically similar to the trend we would like to see, but the statistical significance indicates that there is some ground to make up. Expert evaluations are potentially informative for identifying specific shortcomings and how best to address them.

4.4 Expert Human Evaluations

We performed expert evaluations for the *biography* domain only as we do not have access to weather experts. The four *biography* reviewers are journalists who write short biographies for news archives.

For the *biography* domain, evaluations of the texts were largely similar to the evaluations of the non-expert crowd (76.22 average agreement for the sentence–preference task and 72.95 for the text–understandability task). For example, the **disfluent** ratings were highest for the *DocBase* generations and lowest for the *DocOrig* generations. Also, the **fluent** ratings were highest for the *DocOrig* generations, and while the combined **fluent** and **understandable** are higher for *DocSys* as compared to *DocBase*, the *DocBase* generations had a 10% higher **fluent** score (58.22%) as compared to the *DocSys* **fluent** score (47.97%). Based on notes from the reviewers, the succinctness of the the *DocBase* generations are preferred in some ways as they are in keeping with certain editorial standards. This is further reflected in the sentence preferences being 70% in favor of the *DocBase* generations as compared to the *DocSys* generations (all other sentence comparisons were consistent with the non-expert crowd).

These expert evaluations provide much needed clarity to the NLG process. Overall, our system is generating clearly acceptable texts. Further, there are enough parameters inherent in the system to tune to different domain expectations. This is an encouraging result considering that no experts were involved in the development of the system - a key contrast to many other existing (especially rule-based) NLG systems.

5 Conclusions and Future Work

We have presented a hybrid (template-based and statistical), single–staged NLG system that generates natural sounding texts and is domain–adaptable. Our experiments with both experts and non–experts demonstrate that the system-generated texts are comparable to human–authored texts. The development time to adapt our system to new domains is small compared to other NLG systems; around a week to adapt the system to *weather* and *biography* domains. Most of the development time was spent on creating the domain-specific entity taggers for the *weather* domain. The development time would be reduced to hours if the historical data for a domain is readily available with the corresponding input data.

The main limitation of our system is that it requires significant historical data. Our system does consolidate many traditional components (macro- and micro-planning, lexical choice and aggregation),[5] but the system cannot be applied to the domains with no historical data. The quality and the linguistic variability of the generated text is directly proportional to the amount of historical data available.

We also presented a new automatic metric to evaluate generated texts at document collection level to identify boilerplate texts. This metric computes "syntactic repetitiveness" by counting the number of unique template sequences across the given document collection.

Future work will focus on extending our framework by adding additional features to the model that could improve the quality of the generated text. For example, most NLG pipelines have a separate component responsible for referring expression generation (Krahmer and van Deemter, 2012). While we address the associated concern of data consumption in Section 3.5, we currently do not have any features that would handle referring expression generation. We believe that this is possible by identifying referring expressions in templates and adding features to the model to give higher scores to the templates having relevant referring expressions. We also would like to investigate using all the top-scored templates instead of the highest-scoring template. This would help achieve better syntactic-variability scores by producing more natural-sounding texts.

[5]Lexical choice and aggregation are "handled" insofar as their existence in the historical data.

Acknowledgments

This research is made possible by Thomson Reuters Global Resources (TRGR) with particular thanks to Peter Pircher, Jaclyn Sprtel and Ben Hachey for significant support. Thank you also to Khalid Al-Kofahi for encouragment, Leszek Michalak and Andrew Lipstein for expert evaluations and three anonymous reviewers for constructive feedback.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2012. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods for Natural Language Processing (EMNLP 2010)*, pages 502–512.

Regina Barzilay and Mirella Lapata. 2005. Collective content selection for concept-to-text generation. In *Proceedings of the 2005 Conference on Empirical Methods for Natural Language Processing (EMNLP 2005)*, pages 331–338.

John Bateman and Michael Zock. 2003. Natural language generation. In R. Mitkov, editor, *Oxford Handbook of Computational Linguistics*, Research in Computational Semantics, pages 284–304. Oxford University Press, Oxford.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of NLG systems. In *Proceedings of the European Association for Computational Linguistics (EACL'06)*, pages 313–320.

Anja Belz. 2007. Probabilistic generation of weather forecast texts. In *Proceedings of Human Language Technologies 2007: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'07)*, pages 164–171.

Johan Bos. 2008. Wide-coverage semantic analysis with *Boxer*. In J. Bos and R. Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, volume 1 of *Research in Computational Semantics*, pages 277–286. College Publications.

Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. 2011. Content selection from an ontology-based knowledge base for the generation of football summaries. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, pages 72–81.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*, pages 85–91.

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 for Natural Language Processing (EMNLP 2003)*, pages 2003–2007.

Eduard H. Hovy. 1993. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63:341–385.

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

Thorsten Joachims. 2002. *Learning to Classify Text Using Support Vector Machines*. Kluwer.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT.* Kluwer, Dordrecht.

Colin Kelly, Ann Copestake, and Nikiforos Karamanis. 2009. Investigating content selection for language generation using machine learning. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG)*, pages 130–137.

Ioannis Konstas and Mirella Lapata. 2012. Concept-to-text generation via discriminative reranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 369–378.

Emiel Krahmer and Kees van Deemter. 2012. Computational generation of referring expression: A survey. *Computational Linguistics*, 38(1):173–218.

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 (ACL'98)*, pages 704–710.

Vladimir Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710.

Wei Lu and Hwee Tou Ng. 2011. A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In *Proceedings of the 2011 Conference on Empirical Methods for Natural Language Processing (EMNLP 2011)*, pages 1611–1622.

Wei Lu, Hwee Tou Ng, and Wee Sun Lee. 2009. Natural language generation with tree conditional random fields. In *Proceedings of the 2009 Conference on Empirical Methods for Natural Language Processing (EMNLP 2009)*, pages 400–409.

Kathleen R. McKeown. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press.

Johanna D. Moore and Cecile L. Paris. 1993. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–694.

Kishore Papineni, Slim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 311–318.

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

Ehud Reiter, Somayajulu Sripada, Jim Hunter, and Jin Yu. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167:137–169.

Jacques Robin and Kathy McKeown. 1996. Exmpirically designing and evaluating a new revision-based model for summary generation. *Artificial Intelligence*, 85(1-2).

Somayajulu Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2001. A two-stage model for content determination. In *Proceedings of the 8th European Workshop on Natural Language Generation (ENLG)*, pages 1–8.

Kees van Deemter, Mariët Theune, and Emiel Krahmer. 2005. Real *vs.* template-based natural language generation: a false opposition? *Computational Linguistics*, 31(1):15–24.

Ian Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Techniques with Java Implementation (2nd Ed.)*. Morgan Kaufmann, San Francisco, CA.