Domain-Adaptable Hybrid Generation of RDF Entity Descriptions

Or Biran[*]
n-Join
or@n-join.com

Kathleen McKeown
Columbia University
kathy@cs.columbia.edu

Abstract

RDF ontologies provide structured data on entities in many domains and continue to grow in size and diversity. While they can be useful as a starting point for generating descriptions of entities, they often miss important information about an entity that cannot be captured as simple relations. In addition, generic approaches to generation from RDF cannot capture the unique style and content of specific domains. We describe a framework for hybrid generation of entity descriptions, which combines generation from RDF data with text extracted from a corpus, and extracts unique aspects of the domain from the corpus to create domain-specific generation systems. We show that each component of our approach significantly increases the satisfaction of readers with the text across multiple applications and domains.

1 Introduction

RDF ontologies are a wonderful source for generation: they feature standardized structure, are constantly expending and span many interesting domains. However, generation from RDF introduces two major difficulties. First, RDF contains relationships between entities but often lacks other important information about an entity (e.g., historical background and context) which is hard to capture with simple relations. Second, RDF data spans many domains, and presents the difficulty of handling specific domains in generation.

Generally speaking, there are three approaches: domain-specific approaches (with hand-written or other rules relevant to each domain), which are not scalable; generic approaches (generating in exactly the same way for all domains) which result in unnatural text and miss important content; and domain adaptation, which attempts to automatically transfer an approach from one domain to another.

Our approach aims to leverage the advantages of all three. We present a generic framework of generation *meta-systems* for RDF applications, which uses domain adaptation to create domain-specific systems. Biography and Company Description are examples of applications (an application is the description of RDF entities of a particular type), while Politician and Model are examples of domains within the Biography application.

The reason our framework is able to adapt to new domains automatically is that it relies on hybrid concept-to-text (C2T) and text-to-text (T2T) generation: part of the generated text consists of messages that are created from structured data according to a generic recipe, while another part comes from messages extracted from a domain corpus. In addition, we use existing methods to extract paraphrases and discourse models from the domain corpus, which further refines how text is generated differently for each domain.

2 Related Work

Generation from RDF data is not a new topic. Duboue and McKeown (2003) described a content selection approach for generation from RDF data. Sun and Mellish (2006) present a domain-independent approach for sentence generation from RDF triples. Duma and Klein (2013) propose an architecture for learning end-to-end generation systems from aligned RDF data and sampled generated text. End-to-end concept-to-text systems were proposed by Galanis et al. (2009), Androutsopoulos et al. (2013) and Cimiano et al. (2013), among others. For a survey of the

[*] Work done while at Columbia University

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 306–315,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

history of generation from semantic web data and its difficulties, see (Bouayad-Agha et al., 2014).

Generation *meta-systems* which can be automatically adapted to a new domain have been explored in recent years. Angeli et al. (2010) learn to make decisions about content selection and (separately) template selection from an aligned corpus of database records and text describing them. Kondadadi et al. (2013) describe a framework that learns domain-specific templates, content selection, ordering and template selection from an aligned corpus. Both approaches rely on supervised learning from an aligned corpus of data and sample texts generated from the data, which is a rare resource that does not exist for most domains.

Other recent work has focused on domain adaptation for existing generation systems (as opposed to creating adaptable meta-systems). There has been work on adapting generated text for different user groups (Janarthanam and Lemon, 2010; Gkatzia et al., 2014); adapting summarization systems to new genres (Lloret and Boldrini, 2015); adapting dialog generation systems to new applications (Rieser and Lemon, 2011) and domains (Walker et al., 2007); and parameterizing existing handcrafted systems to increase the range of domains they can handle (Lukin et al., 2015).

In comparison, hybrid C2T-T2T generation is fairly unexplored territory. One recent example is Saldanha et al. (2016), which evaluated two approaches to generating company descriptions - one with Wikipedia structured data, the other utilizing web search results - and determined that the best results were achieved by combining the two. However, the hybrid system in this case was only a concatenation of two independent approaches.

3 Framework Overview

Our approach is a *framework* for creating generation *meta-systems* for specific applications of RDF entity description, such as *biography* and *company description* generation. Each meta-system, in turn, can be automatically adapted to a new *domain* within the application (e.g., the *politician* domain within the *biography* application) with only a simple text corpus, resulting in a *concrete generation system* that is specifically adapted to the domain. The generation system uses hybrid generation, building core messages from RDF data (C2T) and adding domain-specific secondary messages from the text corpus (T2T).

3.1 Semantic Data Structures

Our main data structure is the *Semantic Typed Template (STT)*. An STT is a tuple $\langle V, R, L\rangle$ consisting of a set of vertices labeled with entity types $V = \{v_1, \dots, v_n\}$, a set of edges labeled with relations among the vertices $R = \{r_1, \dots, r_m\}$ and a set of lexical templates $L = \{l_1, \dots, l_k\}$. The lexical templates L are all assumed to be lexicalizations of the semantics of the STT and paraphrases of each other, and must be phrases or sentences (that is, multiple-sentence lexicalizations are not allowed). The STT represents both the meaning and possible realizations of a sentence-level unit of semantics, without directly modeling the meaning in any way other than through the graph embodied in V and R. Instead, the meaning is grounded in the lexical template set.

A *message* is an instance of an STT τ with a concrete set of entities E. The set of types $V(\tau)$ constrains the number and types of entities that are allowed to participate in E, and the set of relations $R(\tau)$ constrains them further (the entities must have the proper relations among them).

3.2 Application Definition

RDF is a framework for organizing data using *triples*. Each triple contains a subject, a predicate and an object. In this paper, we use DBPedia (Auer et al., 2007) as our source of RDF data.

Each RDF application defines a single entity type η: each instance of the application is an entity belonging to this type (that is, there exists a triple such that the subject is the instance entity, the predicate is typeOf and the object is η). In Biography, η = Person, while in Company Description η = Company. In addition, each application defines a domain-differentiating predicate π: in Biography, π = Occupation, while in Company Description π = Industry. π must be chosen so that for each instance of the application, there exists an RDF triple where the subject is the instance entity and the predicate is π.

4 Domain Preparation

Our framework defines each application as a generation *meta-system*: a generic system from which concrete, domain-adapted systems can be created using a text corpus. This section describes the process of domain adaptation.

In this paper, we use Wikipedia as our source for domain corpora (each corpus is the set of Wikipedia articles for all entities of the domain). While it is convenient to select the corpus in this way, there is nothing in the framework that requires the domain corpus to come from Wikipedia.

4.1 Extracting Domain STTs and Messages

Given a new domain corpus, we first extract *definitional sentences*: sentences in the corpus which contain an entity which is an instance of the domain. For example, in the Company Description application, in the Computer Hardware domain, definitional sentences for the entity *Apple* may include “Apple is an American multinational technology company” and “In 1984, Apple launched the Macintosh, the first computer to be sold without a programming language at all”.

To templatize the sentence and find its paraphrases, we use the approach of (Biran et al., 2016). Each definitional sentence is parsed, and NNPs that match an entity in DBPedia become typed slots, resulting in a template and a set of entities that match the slot types. The slot types are determined in two stages - sense disambiguation and hierarchical positioning - both achieved by leveraging the DBPedia ontology in combination with vector representations. We then use the templated paraphrase detection method described in (Biran et al., 2016) to compare the template with existing STTs that match the entities’ types and relations (all of which are known from the RDF ontology). The paraphrasing approach uses sentence-level vector representations to calculate the similarity of the template to all of the existing lexicalizations of an STT. If the template is determined to be a paraphrase for an existing STT, it is added as a new lexicalization; otherwise it is treated as a new STT. This new STT (or the old STT with a new lexicalization) can be used for any entity sets that have the appropriate types and relations.

In addition, we create a domain message from the STT and the entities found in the definitional sentence (effectively making the definitional sentence itself a possible lexicalization of this message, along with any alternative lexicalizations if the STT contains any). This gives us the set of potential secodary messages which we will use in the generation pipeline.

Figure 1 shows an example of this process. Two definitional sentences for the entity are found and templatized, and the first is matched to an existing STT (STT_1) as a paraphrase. The first two lexicalizations of this STT are the default ones, created for all RDF triples as explained in Section 5.1; the third is the template of the definitional sentence. The STT can be used with any matching entity set, but in particular, it is matched to the entity set of the definitional sentence to create domain message 1. The second template cannot be matched to an existing STT, so a new one is created, along with domain message 2.

Entity: Candice Bergen (a model)

Definitional sentences (found in Wikipedia):
- “Candice Bergen was born and raised in Beverly Hills, California”
- “Bergen began her career as a fashion model and appeared on the front cover of Vogue magazine”

Templates:
- [Person] was born and raised in [City]
- [Model] began her career as a fashion model and appeared on the front cover of [Fashion Magazine]

STT_1 (matched through paraphrasing):
$V = \{\text{Person}, \text{City}\}$
$R = \{v_2 \text{ birthPlace } v_1\}$
$L = \{$
“The birth place of [v_1] is [v_2]”,
“[v_1]’s birth place is [v_2]”,
“[v_1] was born and raised in [v_2]”,
$\ldots\}$

Domain message 1:
$STT = STT_1$
$E = \{\text{Candice Bergen}, \text{Beverly Hills}\}$

STT_2 (new, no RDF relation):
$V = \{\text{Model}, \text{Fashion Magazine}\}$
$R = \{\emptyset\}$
$L = \{$ “[v_1] began her career as a fashion model and appeared on the front cover of [v_2]” $\}$

Domain message 2:
$STT = STT_2$
$E = \{\text{Candice Bergen}, \text{Vogue Magazine}\}$

Figure 1: An example of the domain STT and message extraction process.

4.2 Extracting the Discourse Planning Model

A discourse planning model is extracted from the domain corpus as described in (Biran and McKeown, 2015). The model provides prior and transition probabilities for the four top-level Penn Discourse TreeBank (PDTB) (Prasad et al., 2008) discourse relations: *expansion*, *comparison*, *contingency* and *temporal*. These probabilities reflect the discourse style that characterizes the domain, and will be used in Section 5 to determine the ordering of, and relations between, generated messages.

4.3 Extracting the Language Model

The language model used in the realization component of the pipeline is not a typical n-gram model. We are not trying to generate words within a sentence. Instead, we have a set of templates for each message to generate (which corresponds to a sentence or phrase in the final text) and we want to choose one that best fits the context. For this purpose, we define and extract three cross-sentence language models.

The first language model is a cross-sentence model for pairs of words that appear in adjacent sentences. The probability that a word w appears in a sentence if word v appears in the previous sentence, independently of everything else, is

$$P(w|v) = \frac{Count(v,w)}{Count(v)}$$

For the probability of a particular template $\mathcal{T}$ given a selected previous sentence $\mathcal{S}$, we take the average over all word pairs:

$$P_{LM_1}(\mathcal{T}|\mathcal{S}) = \frac{\sum_{(w,v)\in\{\mathcal{T}\times\mathcal{S}\}} P(w|v)}{|\{\mathcal{T}\times\mathcal{S}\}|}$$

The second language model is a POS bigram pair model. It treats POS bigrams as individual words in the first model; in other words, $P_{LM_2}(\mathcal{T}|\mathcal{S})$ is defined in the same way as $P_{LM_1}(\mathcal{T}|\mathcal{S})$, except that w and v stand for POS bigrams (instead of words) in the candidate template and the selected previous sentence, respectively.

The third is a sentence length model. Here we compute the expected length of a sentence $\mathcal{T}$ given the length of the previous sentence $\mathcal{S}$ as

$$E[\#\mathcal{T}|\#\mathcal{S}] = \frac{\sum_{\{\sigma_i:\#\sigma_{i-1}=\#\mathcal{S}\}} \#\sigma_i}{|\{\sigma_i : \#\sigma_{i-1} = \#\mathcal{S}\}|}$$

where $\#\mathcal{S}$ is the length of sentence $\mathcal{S}$ in words. We then smooth this expectation estimate using the estimates of nearby lengths:

$$\tilde{E}[\#\mathcal{T}|\#\mathcal{S}] = \frac{\sum_{i=\#\mathcal{S}-3}^{\#\mathcal{S}+3} E[\#\mathcal{T}|i]}{7}$$

Based on this smoothed expectation, we define the probability of a template $\mathcal{T}$ given a selected previous sentence $\mathcal{S}$:

$$P_{LM_3}(\mathcal{T}|\mathcal{S}) \triangleq \frac{1}{(\#\mathcal{T} - \tilde{E}[\#\mathcal{T}|\#\mathcal{S}])^2}$$

This definition is not intended to have a true probabilistic interpretation, but it preserves an order of likelihood since it increases monotonically as the length of $\mathcal{T}$ approaches the expected values.

These models are used in Section 5 to rank possible templates for a message being generated.

5 Generation

Once a domain has been prepared, we can generate text for any instance in that domain. The generation pipeline contains four components: *core message selection*, *domain message selection*, *discourse planning* and *realization*.

5.1 Core Message Selection

For each instance, we produce one core message from each RDF triple that has the instance's entity as the subject. To create a message from a triple, we first match it to an STT based on the predicate. Each predicate becomes an STT with two entity types (the type of the subject, which is the instance entity, and the type of the object) in V; a single relation between the two types (the predicate) in R; and two simple initial templates in L:

- The (PREDICATE) of $[v_1]$ is $[v_2]$
- $[v_1]$'s (PREDICATE) is $[v_2]$

where (PREDICATE) is replaced with the relevant predicate. Additional templates are then found using paraphrasal template mining as described in the previous section. We also create plural versions for cases where v_2 is a list of entities.

For example, in the biography domain, we create an STT for the *birthDate* predicate with $V = \{person, date\}$; $R = \{v_1 \text{ birthDate } v_2\}$; and an initial template set $L =$ {"The birth date of $[v_1]$ is $[v_2]$", "$[v_1]$'s birth date is $[v_2]$"}. In the preparation stage described in Section 4, L may be expanded with paraphrasal templates found in the corpus, for example "[v_1] was born in [v_2]" (see Figure 1 for an example).

We then create a message that contains the relevant STT and the entities in the triple. In case there are multiple triples with the same subject and predicate but different objects, we create a single message with a plural version of the STT and define the second entity as the list of all objects. We shall refer to the set of core messages as C.

In this paper we separate the content selection problem into two parts. The first (this component) is application-dependent and domain-agnostic, and handles the skeleton or core structure of the generated text; the next component handles additional domain-specific content.

5.2 Domain Message Selection

The set of core messages gives us the core entities which participate in the core messages.

We also have the set of domain messages for the domain which are prepared (extracted from the domain corpus) ahead of time as described in Section 4. The set P of *potential domain messages* for generation is the subset of domain messages which contain the instance entity. In this stage of the pipeline, we select a subset of these potential domain messages to include in the generated text.

To select domain messages, we utilize the energy minimization framework described by Barzilay and Lapata (2005). They describe a formulation that allows efficient optimization of what they call *independent scores* of content units and *link scores* among them through the energy minimization framework. The function to minimize is:

$$\sum_{p \in S} ind_N(p) + \sum_{p \in N} ind_S(p) + \sum_{\substack{p_i \in S \\ p_j \in N}} link(p_i, p_j)$$

where S is the subset of P selected for generation, N is the subset not selected ($P - S = N$), $ind_S(p)$ is p's intrinsic tendency to be selected, $ind_N(p)$ is p's intrinsic tendency to not be selected and $link(p_i, p_j)$ is the dependency score for the link between p_i and p_j. A globally optimal partition of P to S and N can be found in polynomial time by constructing a particular kind of graph and finding a minimal cut partition (Greig et al., 1989).

The base preference of a message p is defined

$$Bp(p) = \begin{cases} |R(\tau(p))| & \text{if } M(p) = E(p) \\ -|E(p) \setminus M(p)|^{\frac{\#L(\tau(p))}{10}} & \text{otherwise} \end{cases}$$

where $M(p)$ is the subset of $E(p)$ - the entities of message p - which contains only entities that participate in at least one relation in $R(\tau(p))$, and $\#L(\tau(p))$ is the average length in words of the templates of the STT $\tau(p)$. This definition results in a positive score for a message where all entities participate in a relation, whose weight is the number of relations it covers; conversely, messages which have entities that do not participate in a relation (*unaccounted entities*), have a negative score which increases in magnitude with the number of unaccounted entities and with the length of the templates realizing them. The intuition is that a long message containing many entities that match no triples is unlikely to be relevant.

Then, we define the individual preference scores $ind(p)$ as an average of the similarity of p to each of the core messages using the Jaccard coefficient as a similarity score:

$$ind(p) = \frac{\sum_{m \in C} J(p, m)}{|C|}$$

Finally, we define $ind_S(p)$ and $ind_N(p)$ as

$$ind_S(p) = \begin{cases} Bp(p) \times ind(p) & \text{if } Bp(p) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$ind_N(p) = \begin{cases} \frac{Bp(p)}{ind(p)} & \text{if } Bp(p) < 0 \\ 0 & \text{otherwise} \end{cases}$$

The link scores $link(p_i, p_j)$ are defined using a type similarity score. In contrast to the individual preference scores, where we maximize the entity overlap with core messages (to avoid including messages with no connection to the core of the text), we should not encourage the domain messages to all share the exact same set of entities. Instead, we focus on a softer semantic similarity: shared entity types. This score enhances the coherence of the generated text (for example, by encouraging a focus on the executives of a company in a particular instance, and on its products in another) but allows a flexible range of messages to be selected. The link score definition is

$$link(p_i, p_j) = \frac{\sum_{(e_i, e_j) \in \{E(p_i) \times E(p_j)\}} typsim(e_i, e_j)}{|\{E(p_i) \times E(p_j)\}|}$$

where

$$typsim(e_i, e_j) = \begin{cases} 1 & \text{if } type(e_i) = type(e_j) \\ 0 & \text{otherwise} \end{cases}$$

Denoting the subset of P selected by this process as $selected(P)$, at the end of this process, we have $M = C \cup selected(P)$ - the full set of messages to be generated.

5.3 Discourse Planning

The discourse planning component transforms the unordered set of messages M into an ordered sequence of paragraphs $\mathcal{P} = (p_1, \dots, p_k)$ where each paragraph p_i is an ordered discourse sequence $p_i = (m_1, r_1, m_2, r_2, \dots, r_{n-1}, m_n)$, where the alternating m_i and r_i are messages and discourse relations, respectively.

First, we calculate the semantic similarity of each pair of messages in M as follows:

$$sim(m_i, m_j) = \cos(\mathcal{V}_{\psi_{m_i}}, \mathcal{V}_{\psi_{m_j}}) link(m_i, m_j)$$

where ψ_{m_i} is the pseudo-sentence of message m_i, constructed by concatenating all of its templates; $\mathcal{V}_{\psi_{m_i}}$ is the vector representing ψ_{m_i}, defined as the geometric mean of the vectors of all words participating in ψ_{m_i} (the word vectors are traditional context vectors extracted from Gigaword with a window of 5 words); and $link(m_i, m_j)$ is defined as above. Essentially, this is a combination of the entity type-based semantic similarity and the distributional similarity of the lexicalizations.

We use single-linkage agglomerative clustering (with a stopping criteria of $sim(m_i, m_j) \leq 0.05$) to group the messages into semantic groups of messages that are similar in topic. Then, within each semantic group, we find potential discourse relations for each pair of messages:

1. If the STTs of m_i and m_j are the same but they have no entities in common then there is a potential *comparison* relation between them
2. If $J(m_i, m_j) \geq 0.5$ then there is a potential *expansion* relation between them
3. Manually annotated relations for 20 specific pairs of RDF predicates, e.g. *birthPlace* and *residence* may have a temporal or a comparison relation between them
4. All message pairs can have a *norel* relation

Next, we use the discourse model extracted from the domain corpus to generate a discourse sequence. In order to make sure entity coherence is taken into account when choosing the ordering in addition to discourse coherence, we augment the probabilities coming from the discourse model $P_{\mathcal{D}}(r_i|R_{i-1})$, where R_{i-1} is the sequence of relations chosen so far, with the entity coherence score $J(m_i, m_{i-1})$, so that the probability of a relation between two messages is given by

$$P(r_i|R_{i-1}, m_i, m_{i-1}) = P_{\mathcal{D}}(r_i|R_{i-1}) J(m_i, m_{i-1})$$

The discourse sequence is created stochastically using these probabilities as described in (Biran and McKeown, 2015). Then, we break the discourse sequence into paragraphs that do not contain *norel* relations. Concatenating all of the paragraphs built from the discourse sequences of all semantic groups, we have an unordered set of paragraphs $\mathcal{P}$, where each p_i is an ordered discourse sequence of messages and relations.

To order the paragraphs, we use the following importance score:

$$imp(p_i) = \frac{\sum_{m \in p_i} |\{e | e \in E(m)\}| Bp(m)}{|p_i|}$$

which is the average number of entities in a message of p_i, weighted by the base preference score $Bp(m)$. The paragraphs are then sorted in decreasing order using this score, so that the paragraphs containing the most important messages tend to appear earlier in the text.

5.4 Realization

At this stage, we have the ordered set of paragraphs $\mathcal{P}$ to be realized. To generate a paragraph, we select a template for each message and then select a discourse connective, or choose not to use one, for each discourse relation.

Selecting a template is done using the three language models prepared ahead of time, as described in Section 4. We build a ranker from each model, and choose the template from $\{l \in L(\tau(m))\}$ that maximizes the the sum of ranks given the previously realized sentence (in the paragraph) s:

$$\hat{l} = \underset{l \in L(\tau(m))}{\operatorname{argmax}} \sum_{i=1}^{3} rank\big(P_{LM_i}(l|s)\big)$$

Once the template is chosen, we fill the slots with the entities $E(m)$ to make it a sentence.

At this point we have the final lexical form of the message, and the last task is to link it with the previous sentence. We have a small set of discourse connective templates for each one of the 4 class-level PDTB relations (for example, "m_i. However, m_j" is one of the templates for the *comparison* relation), and we know the relation between the message and the previous message. We randomly select a connective, with a 50% chance of having no connective and a uniform distribution among the connectives for the relation, but avoid using connectives for sentence pairs that are together larger than 40 words.

At the end of this step, all paragraphs are generated with lexicalized sentences and connectives.

6 Evaluation

To evaluate our RDF applications we conducted a crowd-sourced human experiment using texts generated from four domains in two applications: Biographies of *Politicians* and *Models*, and Company Descriptions of *Automobile Manufacturers* and *Video Game Developers*. We picked 100 instances from each domain of each application, for a total of 400 (we picked the instances that had the most RDF triples in each domain). Then, we generated 4 versions for each instance:

1. A full-system version
2. A version that excludes paraphrase detection (so core messages only had the two manually-created templates, and domain messages only had a single template each)
3. A version that excludes the discourse model (so discourse planning was done using only entity coherence scores)
4. A baseline version that has no domain adaptation at all and is fully C2T instead of hybrid (i.e., only core messages were generated, without any extracted domain messages)

Using these 4 versions, we devised 3 questions for each instance. In each question, the annotator saw two texts about the same entity - the full system version, and one of the other three versions - and was asked which is better (with an option of saying they are equal), along several criteria. The questions were presented in random order and the systems were anonymized. We showed each question to three annotators and used the majority vote, throwing out results where there was total disagreement between the annotators, which happened 12% of the time for the baseline version and $17 - 21\%$ of the time for the other variants.

The questions included four criteria: the *content* of the text (information relevance); the *ordering* of the sentences and paragraphs; the *style* of the text (how human-like it is); and the *overall* satisfiability of the text as a description of the person/company in question.

We show the results of the experiment in Table 1. The results in this table are for both applications and all four domains. Each comparison (e.g., "No Hybrid VS Full System" shows the breakdown of preference by annotators when they were shown texts generated by the two variants: how many (in percentage) preferred the baseline system (e.g. No Hybrid), how many preferred the full system, and how many thought they were equal. We also show the *winning difference* between the two systems, i.e. those who thought that the full system was better than the baseline minus those who thought the opposite, and we measure statistical significance on these differences. Statistically significant results are marked with a dagger.

6.1 Discussion

The most striking result of Table 1 is that the full system is overwhelmingly favored by annotators over the non-hybrid baseline, with a $32\% - 46\%$ lead in all categories. This result, more than anything, shows the value of our framework and the hybrid approach. The full system was particularly better than this baseline in *content*, which is generally expected since it by definition contains less content than the full system (it only generates the core messages); note, however, that this result suggests that the extracted and selected messages are *relevant* and enhance the reader's satisfaction with the text. The baseline (which, in addition to not using extracted domain messages, also does not use the extracted paraphrasal templates and discourse model) also loses heavily to the full system in *ordering* and *style*, as well as overall. In all criteria, the percentage of annotators who thought the texts were equally good was low ($11\% - 20\%$), suggesting that the difference was very visible.

While the effect of removing a single component is not as dramatic as removing both in addition to the domain messages, it is clearly visible in the preferences of Table 1. Both reduced versions (*No Paraphrases* and *No Discourse Model*) lose to the full system in every criteria, often in double digits. The more meaningful component

	Preference	Content	Ordering	Style	Overall
No Hybrid VS Full System	No Hybrid	20%	27%	24%	22%
	Equal	14%	11%	20%	14%
	Full System	66%	62%	56%	64%
	Full - baseline win diff.	**46%** †	**35%** †	**32%** †	**42%** †
No Paraphrases VS Full System	No Paraphrases	29%	33%	29%	30%
	Equal	31%	26%	28%	27%
	Full System	40%	41%	43%	43%
	Full - baseline win diff.	**11%** †	**8%** †	**14%** †	**13%** †
No Discourse Model VS Full System	No Discourse Model	33%	34%	32%	34%
	Equal	30%	22%	26%	23%
	Full System	37%	44%	42%	43%
	Full - baseline win diff.	**4%**	**10%** †	**10%**	**9%** †

Table 1: Preferences, with different criteria, given by the human annotators when presented with two versions - the full system VS each of the baseline versions. Statistically significant winning differences are marked with a dagger.

appears to be the paraphrases: the *No Paraphrases* version loses to the full system more heavily than *No Discourse* in *content*, *style* and *overall*. This result is not surprising since paraphrases have a dramatic effect on the text itself (they change the templates that are used to convey information, enhance the diversity of the text and may merge messages that are duplicates), and it suggests that the paraphrases we find are generally more satisfying than the default. It is also not surprising that the *No Discourse Model* variant loses most on ordering. While the difference is not as dramatic here, it is statistically significant and shows that our extracted domain-specific discourse model produces a more satisfying ordering of the text.

6.2 Examples

Figure 2 shows the output of the biography for politician *Marine Le Pen* of the full system and the non-hybrid baseline. To show the contributions of different components, we mark sentences generated from extracted domain messages in **bold**, and sentences generated from core messages using an extracted paraphrase in *italics*. Sentences in unmarked typeface are those that were generated from core messages using a default template.

The two variants make clear the main advantage of the full system: it simply has more content. The full output contains six sentences (messages) more than the baseline, which are clearly relevant to the biography. The entire last paragraph, concerned with Le Pen's policies and positions - an important part of a politician's biography - is missing from the baseline. These messages were extracted from the corpus and show the power of the hybrid approach. In addition to the final paragraph, two extracted messages are included which are concerned with Le Pen's controversial history, and together with the RDF-derived message about her offices, they comprise a paragraph generally about her political background. This is typical of the way that extracted messages contribute to the organization of the text in addition to the content: in the baseline version, the offices message is lumped together with messages about her background in general (alma mater, birth date, religion, partner etc). It demonstrates how the full system consistently outperforms the baseline in the *ordering* and *style* criteria, in addition to *content* and *overall*.

Figure 3 shows the output of the company description for video game developer *Taito Corporation* of the full system and the no-paraphrases variant. In this case the two outputs contain exactly the same information and have almost the same organization of the text. The way in which the text is realized, however, is very different in the last paragraph. The full system realizes four of the six messages in that paragraph using extracted templates, and merges two messages into a single template in one case ("Taito Corporation was founded in 1953 by Michael Kogan", instead of the two sentences in the no-paraphrases baseline). The single-sentence messages also look better, e.g. "Taito Corporation has around 662 employees" instead of the awkward-sounding "Taito

Full system output:

Marine Le Pen’s birth places are Neuilly-sur-Seine and France. Marine Le Pen’s residences are Millas, Hénin-Beaumont and Saint-Cloud.

The birth name of Marine Le Pen is Marion Anne Perrine Le Pen. Marine Le Pen’s offices are Leader of the National Front, Municipal Councillor, Member of the European Parliament and Regional Councillor. **Marine Le Pen’s ups and downs in the political arena follow those of the National Front at the time. Marine Le Pen stirred up controversy during the internal campaign.**

The homepage of Marine Le Pen is http://www.marinelepen.fr/.

The alma mater of Marine Le Pen is Panthéon-Assas University. Marine Le Pen’s birth date was 1968-08-05. Marine Le Pen’s religion is Catholic Church. Marine Le Pen’s occupation is Politician. Marine Le Pen’s partner is Louis Aliot.

Marine Le Pen regularly denounces sharp rises in energy prices which has “harmful consequences on the purchasing power of the working and middle-class families”. Marine Le Pen denounces the current corporate tax as “a crying injustice”. Marine Le Pen advocates to “vote for the abolition of the law enabling the regularization of the illegal immigrants”. Marine Le Pen seeks to establish a moratorium on legal immigration.

Baseline output:

Marine Le Pen’s party is National Front. Marine Le Pen’s occupation is Politician. Marine Le Pen’s homepage is http://www.marinelepen.fr/. Marine Le Pen’s offices are Leader of the National Front, Municipal Councillor, Member of the European Parliament and Regional Councillor. Marine Le Pen’s birth name is Marion Anne Perrine Le Pen. Marine Le Pen’s religion is Catholic Church. Marine Le Pen’s alma mater is Panthéon-Assas University. Marine Le Pen’s birth date was 1968-08-05. Marine Le Pen’s partner is Louis Aliot.

The birth places of Marine Le Pen are Neuilly-sur-Seine and France. Marine Le Pen’s residences are Millas, H’enin-Beaumont and Saint-Cloud.

Figure 2: Output for *Marine Le Pen*.

Full system output:

The homepage of Taito Corporation is http://www.taito.com.

The products of Taito Corporation are Lufia, Bubble Bobble, Cooking Mama, Space Invaders, Chase H.Q., Gun Fight and Puzzle Bobble.

Taito Corporation was founded in 1953 by Michael Kogan. Taito Corporation has around 662 employees. Taito Corporation’s location is Shibuya, Tokyo, Japan. **Taito Corporation currently has a subsidiary in Beijing, China.** *Taito Corporation was merged with “Square Enix”.*

No-paraphrases output:

Taito Corporation’s homepage is http://www.taito.com.

The products of Taito Corporation are Lufia, Bubble Bobble, Cooking Mama, Space Invaders, Chase H.Q., Gun Fight and Puzzle Bobble.

Taito Corporation’s founding year is 1953. The founder of Taito Corporation is Michael Kogan. Taito Corporation’s owner is Square Enix. **Taito Corporation currently has a subsidiary in Beijing, China.** Taito Corporation’s location is Shibuya, Tokyo, Japan. Taito Corporation’s number of employees is 662.

Figure 3: Output for *Taito Corporation*.

Corporation’s number of employees is 662”.

7 Conclusion

We introduced a framework for creating hybrid concept-to-text and text-to-text generation systems that produce descriptions of RDF entities, and can be automatically adapted to a new domain with only a simple text corpus. We showed through a human evaluation that both the hybrid approach and domain adaptation result in significantly more satisfying descriptions, and that individual methods of domain adaptation help with the criteria we expect them to (i.e., finding paraphrases helps with content and style while an extracted discourse model helps with ordering). The code for this framework is available at www.cs.columbia.edu/~orb/hygen/.

References

Ion Androutsopoulos, Gerasimos Lampouras, and Dimitrios Galanis. 2013. Generating natural language descriptions from owl ontologies: the naturalowl system. *Journal of Artificial Intelligence Research*, pages 671–715.

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP ’10, pages 502–512, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC’07/ASWC’07, pages 722–735, Berlin, Heidelberg. Springer-Verlag.

Regina Barzilay and Mirella Lapata. 2005. Collective content selection for concept-to-text generation. In *Proceedings of the HLT/EMNLP*, pages 331–338, Vancouver.

Or Biran, Terra Blevins, and Kathleen McKeown. 2016. Mining paraphrasal typed templates from a plain text corpus. In *Proceedings of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics.

Or Biran and Kathleen McKeown. 2015. Discourse planning with an n-gram model of relations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1973–1977, Lisbon, Portugal. Association for Computational Linguistics.

Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. 2014. Natural language generation in the context of the semantic web. *Semantic Web*, 5(6):493–513.

Philipp Cimiano, Janna Lüker, David Nagel, and Christina Unger. 2013. Exploiting ontology lexica for generating natural language texts from rdf data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 10–19, Sofia, Bulgaria. Association for Computational Linguistics.

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.

Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 83–94. ASSOC COMPUTATIONAL LINGUISTICS-ACL.

Dimitrios Galanis, George Karakatsiotis, Gerasimos Lampouras, and Ion Androutsopoulos. 2009. An open-source natural language generator for owl ontologies and its use in protÉgÉ and second life. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, EACL '09, pages 17–20, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dimitra Gkatzia, Helen Hastie, and Oliver Lemon. 2014. Multi-adaptive natural language generation using principal component regression. In *Proceedings of the International Natural Language Generation (INLG)*.

D. M. Greig, B. T. Porteous, and A. H. Seheult. 1989. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, 51(2):271–279.

Srinivasan Janarthanam and Oliver Lemon. 2010. Adaptive referring expression generation in spoken dialogue systems: Evaluation with real users. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '10, pages 124–131, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ravi Kondadadi, Blake Howald, and Frank Schilder. 2013. A statistical NLG framework for aggregated planning and realization. In *ACL*, pages 1406–1415. The Association for Computer Linguistics.

E. Lloret and E. Boldrini. 2015. Multi-genre summarization: Approach, potentials and challenges. In *eChallenges e-2015 Conference*, pages 1–9.

Stephanie Lukin, Lena Reed, and Marilyn Walker. 2015. Generating sentence planning variations for story telling. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 188–197, Prague, Czech Republic. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *In Proceedings of LREC*.

Verena Rieser and Oliver Lemon. 2011. Learning and evaluation of dialogue strategies for new applications: Empirical methods for optimization from small data sets. *Comput. Linguist.*, 37(1):153–196.

Gavin Saldanha, Or Biran, Kathleen McKeown, and Alfio Gliozzo. 2016. An entity-focused approach to generating company descriptions. In *Proceedings of the Association for Computational Linguistics (ACL)*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiantang Sun and Chris Mellish. 2006. Domain independent sentence generation from rdf representations for the semantic web. In *Combined Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems, European Conference on AI*.

Marilyn Walker, Amanda Stent, François Mairesse, and Rashmi Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. *J. Artif. Int. Res.*, 30(1):413–456.