Learning to generate one-sentence biographies from Wikidata

Andrew Chisholm
University of Sydney
Sydney, Australia
andy.chisholm.89@gmail.com

Will Radford
Hugo Australia
Sydney, Australia
wradford@hugo.ai

Ben Hachey
Hugo Australia
Sydney, Australia
bhachey@hugo.ai

Abstract

We investigate the generation of one-sentence Wikipedia biographies from facts derived from Wikidata slot-value pairs. We train a recurrent neural network sequence-to-sequence model with attention to select facts and generate textual summaries. Our model incorporates a novel secondary objective that helps ensure it generates sentences that contain the input facts. The model achieves a BLEU score of 41, improving significantly upon the vanilla sequence-to-sequence model and scoring roughly twice that of a simple template baseline. Human preference evaluation suggests the model is nearly as good as the Wikipedia reference. Manual analysis explores content selection, suggesting the model can trade the ability to infer knowledge against the risk of hallucinating incorrect information.

1 Introduction

Despite massive effort, Wikipedia and other collaborative knowledge bases (KBs) have coverage and quality problems. Popular topics are covered in great detail, but there is a long tail of specialist topics with little or no text. Other text can be incorrect, whether by accident or vandalism. We report on the task of generating textual summaries for people, mapping slot-value facts to one-sentence encyclopaedic biographies. In addition to initialising stub articles with only structured data, the resulting model could be used to improve consistency and accuracy of existing articles. Figure 1 shows a Wikidata entry for `Mathias Tuomi`, with fact keys and values flattened into a sequence, and the first sentence from his Wikipedia article. Some values are in the text, others are missing (e.g. male) or expressed differently (e.g. dates).

```
TITLE mathias tuomi SEX_OR_GENDER
male DATE_OF_BIRTH 1985-09-03
OCCUPATION squash player
CITIZENSHIP finland
```

Figure 1: Example Wikidata facts encoded as a flat input string. The first sentence of the Wikipedia article reads: Mathias Tuomi, (born September 30, 1985 in Espoo) is a professional squash player who represents Finland.

We treat this *knowlege-to-text* task like translation, using a recurrent neural network (RNN) sequence-to-sequence model (Sutskever et al., 2014) that learns to select and realise the most salient facts as text. This includes an attention mechanism to focus generation on specific facts, a shared vocabulary over input and output, and a multi-task autoencoding objective for the complementary extraction task. We create a reference dataset comprising more than 400,000 knowledge-text pairs, handling the 15 most frequent slots. We also describe a simple template baseline for comparison on BLEU and crowd-sourced human preference judgements over a heldout TEST set.

Our model obtains a BLEU score of 41.0, compared to 33.1 without the autoencoder and 21.1 for the template baseline. In a crowdsourced preference evaluation, the model outperforms the baseline and is preferred 40% of the time to the Wikipedia reference. Manual analysis of content selection suggests that the model can infer knowledge but also makes mistakes, and that the autoencoding objective encourages the model to select more facts without increasing sentence length. The task formulation and models are a foundation for text completion and consistency in KBs.

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 633–642,
Valencia, Spain, April 3-7, 2017.

2 Background

RNN sequence-to-sequence models (Sutskever et al., 2014) have driven various recent advances in natural language understanding. While initial work focused on problems that were sequences of the same units, such as translating a sequence of words from one language to another, other work been able to use these models by *coercing* different structures into sequences, e.g., flattening trees for parsing (Vinyals et al., 2015), predicting span types and lengths over byte input (Gillick et al., 2016) or flattening logical forms for semantic parsing (Xiao et al., 2016).

RNNs have also been used successfully in *knowledge-to-text* tasks for human-facing systems, e.g., generating conversational responses (Vinyals and Le, 2015), abstractive summarisation (Rush et al., 2015). Recurrent LSTM models have been used with some success to generate text that completely expresses a set of facts: restaurant recommendation text from dialogue acts (Wen et al., 2015), weather reports from sensor data and sports commentary from on-field events (Mei et al., 2015). Similarly, we learn an end-to-end model trained over key-value facts by flattening them into a sequence.

Choosing the salient and consistent set of facts to include in generated output is also difficult. Recent work explores unsupervised autoencoding objectives in sequence-to-sequence models, improving both text classification as a pretraining step (Dai and Le, 2015) and translation as a multitask objective (Luong et al., 2016). Our work explores an autoencoding objective which selects content as it generates by constraining the text output sequence to be predictive of the input.

Biographic summarisation has been extensively researched and is often approached as a sequence of subtasks (Schiffman et al., 2001). A version of the task was featured in the Document Understanding Conference in 2004 (Blair-Goldensohn et al., 2004) and other work learns policies for content selection without generating text (Duboue and McKeown, 2003; Zhang et al., 2012; Cheng et al., 2015). While pipeline components can be individually useful, integrating selection and generation allows the model to exploit the interaction between them.

KBs have been used to investigate the interaction between structured facts and unstructured text. Generating textual templates that are filled by structured data is a common approach and has been used for conversational text (Han et al., 2015) and biographical text generation (Duma and Klein, 2013). Wikipedia has also been a popular resource for studying biography, including sentence harvesting and ordering (Biadsy et al., 2008), unsupervised discovery of distinct sequences of life events (Bamman and Smith, 2014) and fact extraction from text (Garera and Yarowsky, 2009). There has also been substantial work in generating from other structured KBs using template induction (Kondadadi et al., 2013), semantic web techniques (Power and Third, 2010), tree adjoining grammars (Gyawali and Gardent, 2014), probabilistic context free grammars (Konstas and Lapata, 2012) and probabilistic models that jointly select and realise content (Angeli et al., 2010).

Lebret et al. (2016) present the closest work to ours with a similar task using Wikipedia infoboxes in place of Wikidata. They condition an attentional neural language model (NLM) on local and global properties of infobox tables, including *copy actions* that allow wholesale insertion of values into generated text. They use 723k sentences from Wikipedia articles with 403k lower-cased words mapping to 1,740 distinct facts. They compare to a 5-gram language-model with copy actions, and find that the NLM has higher BLEU and lower perplexity than their baseline. In contrast, we utilise a deep recurrent model for input encoding, minimal slot value templating and greedy output decoding. We also explore a novel autoencoding objective that measures whether input facts can be re-created from the generated sentence.

Evaluating generated text is challenging and no one metric seems appropriate to measure overall performance. Lebret et al. (2016) report BLEU scores (Papineni et al., 2002) which calculate the n-gram overlap between text produced by the system with respect to a human-written reference. Summarisation evaluations have concentrated on the content that is included in the summary, with semantic content typically extracted manually for comparison (Lin and Hovy, 2003; Nenkova and Passonneau, 2004). We draw from summarisation and generation to formulate a comprehensive evaluation based on automated metrics and human validation. Our final system comparison follows Kondadadi et al. (2013) in running a crowd task to collect pairwise preferences for evaluating and comparing both systems and references.

Fact	Count	%
TITLE (name)	1,011,682	98
SEX_OR_GENDER	1,007,575	0
DATE_OF_BIRTH	817,942	88
OCCUPATION	720,080	67
CITIZENSHIP	663,707	52
DATE_OF_DEATH	346,168	86
PLACE_OF_BIRTH	298,374	25
EDUCATED_AT	141,334	32
SPORTS_TEAM	108,222	29
PLACE_OF_DEATH	107,188	17
POSITION_HELD	87,656	75
PARICIPANT_OF	77,795	23
POLITICAL_PARTY	74,371	49
AWARD_RECEIVED	67,930	44
SPORT	36,950	72

Table 1: The top fifteen slots across entities used for input, and the % of time the value is a substring in the entity's first sentence.

3 Task and Data

We formulate the one-sentence biography generation task as shown in Figure 1. Input is a flat string representation of the structured data from the KB, comprising slot-value pairs (the subject being the topic of the KB record, e.g., Mathias Tuomi), ordered by slot frequency from most to least common. Output is a biography string describing the salient information in one sentence.

We validate the task and evaluation using a closely-aligned set of resources: Wikipedia and Wikidata. In addition to the KB maintenance issues discussed in the introduction, Wikipedia first sentences are of particular interest because they are clear and concise biographical summaries. These could be applied to entities outside Wikipedia for which one can obtain comparable parallel structured/textual data, e.g., movie summaries from IMDb, resume overviews from LinkedIn, product descriptions from Amazon.

We use snapshots of Wikidata (2015/07/13) and Wikipedia (2015/10/02) and batch process them to extract instances for learning. We select all entities that are `INSTANCE_OF human` in Wikidata. We then use `sitelinks` to identify each entity's Wikipedia article text and NLTK (Bird et al., 2009) to tokenize and extract the lower-cased first sentence. This results in 1,268,515 raw knowledge-text pairs. The summary sentences can be long and the most frequent length is 21 tokens. We filter to only include those between the 10th and 90th percentiles: 10 and 37 tokens. We split this collection into TRAIN, DEV and TEST collections with 80%, 10% and 10% of instances allocated respectively. Given the large variety of slots which may exist for an entity, we restrict the set of slots used to the top-15 by occurrence frequency. This criteria covers 72.8% of all facts. Table 1 shows the distribution of fact slots in the structured data and the percentage of time tokens from a fact value occur in the corresponding Wikipedia summary.

Additionally, some Wikidata entities remain underpopulated and do not contain sufficient facts to reconstruct a text summary. We control for this information mismatch by limiting our dataset to include only instances with at least 6 facts present. The final dataset includes 401,742 TRAIN, 50,017 DEV and 50,030 TEST instances. Of these instances, 95% contain 6 to 8 slot values while 0.1% contain the maximum of 10 slots. 51% of unique slot-value pairs expressed in TEST and DEV are not observed in TRAIN so generalisation of slot usage is required for the task. The KB facts give us an opportunity to measure the correctness of the generated text in a more precise way than text-to-text tasks. We use this for analysis in Section 7.3, driving insight into system characteristics and implications for use.

3.1 Task complexity

Wikipedia first sentences exhibit a relatively narrow domain of language in comparison to other generation tasks such as translation. As such, it is not clear how complex the generation task is, and we first try to use perplexity to describe this.

We train both RNN models until DEV perplexity stops improving. Our basic sequence-to-sequence model (S2S) reaches perplexity of 2.82 on TRAIN and 2.92 on DEV after 15,000 batches of stochastic gradient descent. The autoencoding sequence-to-sequence model (S2S+AE) takes longer to fit, but reaches a lower minimum perplexity of 2.39 on TRAIN and 2.51 on DEV after 25,000 batches.

To help ground perplexity numbers and understand the complexity of sentence biographies we train a benchmark language model and evaluate perplexity on DEV. Following Lebret et al. (2016), we build Kneser-Ney smoothed 5-gram language models using the KenLM toolkit (Heafield, 2011).

Table 2 lists perplexity numbers for the benchmark LM models with different templating

Templates	DEV
None	29.8
Title	14.5
Full	10.1

Table 2: Language model perplexity across templated datasets.

schemes on DEV. We observe decreasing perplexity for data with greater fact value templating. TITLE indicates templating of entity names only, while FULL indicates templating of all fact values by token index as described in Lebret et al. (2016). This shows that templating is an effective way to reduce the sparsity of a task, and that titles account for a large component of this.

Although Lebret et al. (2016) evaluate on a different dataset, we are able to draw some comparisons given the similarity of our task. On their data, the benchmark LM baseline achieves a similar perplexity of 10.5 to ours when following their templating scheme on our dataset - suggesting both samples are of comparable complexity.

4 Model

We model the task as a sequence-to-sequence learning problem. In this setting, a variable length input sequence of entity facts is encoded by a multi-layer RNN into a fixed-length distributed representation. This input representation is then fed into a separate decoder network which estimates a distribution over tokens as output. During training, parameters for both the encoder and decoder networks are optimized to maximize the likelihood of a summary sequence given an observed fact sequence.

Our setting differs from the translation task in that the input is a sequence representation of structured data rather than natural human language. As described above in Section 3, we map Wikidata facts to a sequence of tokens that serves as input to the model as illustrated at the top of Figure 2. Experiments below demonstrate that this is sufficient for end-to-end learning in the generation task addressed here. To generate summaries, our model must both select relevant content and transform it into a well formed sentence. The decoder network includes an attention mechanism (Vinyals et al., 2015) to help facilitate accurate content selection. This allows the network to focus on different parts of the input sequence during inference.

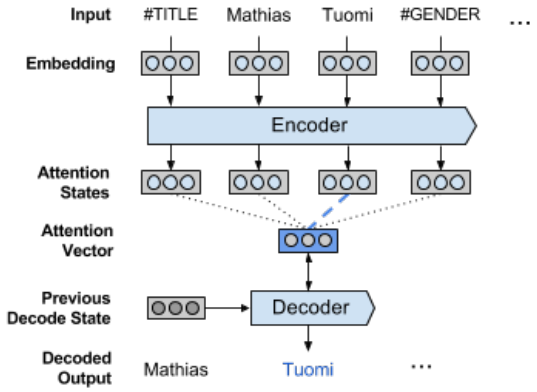

Figure 2: Sequence-to-sequence translation from linearized facts to text.

4.1 Sequence-to-sequence model (S2S)

To generate language, we seed the decoder network with the output of the encoder and a designated GO token. We then generate symbols greedily, taking the most likely output token from the decoder at each step given the preceding sequence until an EOS token is produced. This approach follows (Sutskever et al., 2014) who demonstrate a larger model with greedy sequence inference performs comparably to beam search. In contrast to translation, we might expect good performance on the summarization task where output summary sequences tend to be well structured and often formulaic. Additionally, we expect a partially-shared language across input and output. To exploit this, we use a tied embedding space, which allows both the encoder and decoder networks to share information about word meaning between fact values and output tokens.

Our model uses a 3-layer stacked Gated Recurrent Unit RNN for both encoding and decoding, implemented using TensorFlow.[1] We limit the shared vocabulary to 100,000 tokens with 256 dimensions for each token embedding and hidden layer. Less common tokens are marked as UNK, or unknown. To account for the long tail of entity names, we replace matches of title tokens with templated copy actions (e.g. TITLE0 TITLE1...). These template are then filled after generation, as well as any initial unknown tokens in the output, which we fill with the first title token. We learn using minibatch Stochastic Gradient Descent with a batch size of 64 and a fixed learning rate of 0.5.

[1] https://www.tensorflow.org, v0.8.

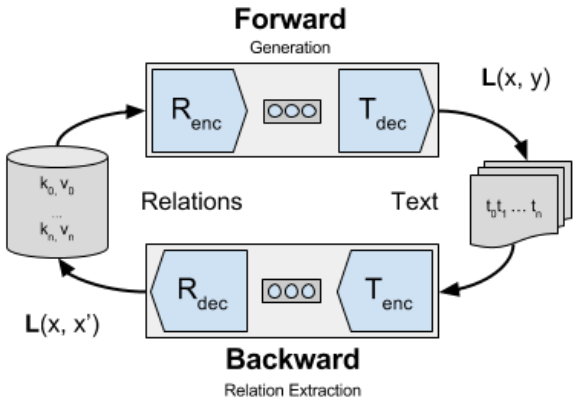

Figure 3: Sequence-to-sequence autoencoder.

4.2 S2S with autoencoding (S2S+AE)

One challenge for vanilla sequence-to-sequence models in this setting is the lack of a mechanism for constraining output sequences to only express those facts present in the data. Given a fact extraction oracle, we might compare facts expressed in the output sequence with those of the input and appropriately adjust the loss for each instance. While a forward-only model is only constrained to generate text sequences predicted by the facts, an autoencoding model is additionally constrained to generate text predictive of the input facts. In place of this ideal setting, we introduce a second sequence-to-sequence model which runs in reverse - re-encoding the text output sequence of the forward model into facts.

This closed-loop model is detailed in Figure 3. The resulting network is trained end-to-end to minimize both the input-to-output sequence loss $L(x, y)$ and output-to-input reconstruction loss $L(x, x')$. While gradients cannot propagate through the greedy forward decode step, shared parameters between the forward and backward network are fit to both tasks. To generate language at test time, the backward network does not need to be evaluated.

5 Experimental methodology

The evaluation suite here includes standard baselines for comparison, automated metrics for learning, human judgement for evaluation and detailed analysis for diagnostics. While each are individually useful, their combination gives a comprehensive analysis of a complex problem space.

5.1 Benchmarks

WIKI We use the first sentence from Wikipedia both as a gold standard reference for evaluating generated sentences, and as an upper bound in human preference evaluation.

BASE Template-based systems are strong baselines, especially in human evaluation. While output may be stilted, the corresponding consistency can be an asset when consistency is important. We induce common patterns from the TRAIN set, replacing full matches of values with their slot and choosing randomly on ties. Multiple non-fact tokens are collapsed to a single symbol. A small sample of the most frequent patterns were manually examined to produce templates, roughly expressed as: TITLE, known as GIVEN_NAME, (born DATE_OF_BIRTH in PLACE_OF_BIRTH; died DATE_OF_DEATH in PLACE_OF_DEATH) is an POSITION_HELD and OCCUPATION from CITIZENSHIP, with some sensible back-offs where slots are not present, and rules for determiner agreement and is versus was where a death date is present. For example, ollie freckingham (born 12 november 1988) is a cricketer from the united kingdom. In total, there are 48 possible template variations.

5.2 Metrics

BLEU We also report BLEU n-gram overlap with respect to the reference Wikipedia summary. With a large dev/test sets (10,000 sentences here), BLEU is a reasonable evaluation of generated content. However, it does not give an indication of well-formedness or readability. Thus we complement BLEU with a human preference evaluation.

Human preference We use crowd-sourced judgements to evaluate the relative quality of generated sentences and the reference Wikipedia first sentence. We obtain pairwise judgements, showing output from two different systems to crowd workers and asking each to give their binary preference. The system name mappings are anonymized and ordered pseudo-randomly. We request 3 judgements and dynamically increase this until we reach at least 70% agreement or a maximum of 5 judgements. We use CrowdFlower[2] to collect judgements at the cost of 31 USD for all 6 pairwise combinations over 82

[2] http://www.crowdflower.com

	DEV	TEST
Base	21.3	21.1
S2S	32.5	33.1
S2S+AE	40.5	41.0

Table 3: BLEU scores for each hypothesis against the Wikipedia reference

S2S+AE	BASE	S2S	
60%	61%*	87%**	WIKI
	62%*	77%**	S2S+AE
		65%**	BASE

Table 4: Percentage of entities for which human judges preferred the row system to the column system. E.g., S2S+AE summaries are preferred to BASE for 62% of sample entities.

randomly selected entities. 67 workers contributed judgements to the test data task, each providing no more than 50 responses. We use the majority preference for each comparison. The CrowdFlower agreement is 80.7%, indicating that roughly 4 of 5 votes agree on average.

5.3 Analysis of content selection

Finally, no system is perfect, and it can be challenging to understand the inherent difficulty of the problem space and the limitations of a system. Due to the limitations of the evaluation metrics mentioned above, we propose that manual annotation is important and still required for qualitative analysis to guide system improvement. The structured data in knowledge-to-text tasks allows us, if we can identify expressions of facts in text, cases where facts have been omitted, incorrectly mentioned, or expressed differently.

6 Results

6.1 Comparison against Wikipedia reference

Table 3 shows BLEU scores calculated over 10,000 entities sampled from DEV and TEST using the Wikipedia sentence as a single reference, using uniform weights for 1- to 4-grams, and padding sentences with fewer than 4 tokens. Scores are similar across DEV and TEST, indicating that the samples are of comparable difficulty. We evaluate significance using bootstrapped resampling with 1,000 samples. Each system result lies outside the 95% confidence intervals of other systems. BASE has reasonable scores at 21, with S2S higher at around 32, indicating that the model is at least able to generate closer text than the baseline. S2S+AE scores higher still at around 41, roughly double the baseline scores, indicating that the autoencoder is indeed able to constrain the model to generate better text.

6.2 Human preference evaluation

Table 4 shows the results of our human evaluation over 82 entities sampled from TEST. For each pair of systems, we show the percentage of entities where the crowd preferred A over B. Significant differences are annotated with $*$ and $**$ for p values < 0.05 and 0.01 using a one-way χ^2 test. WIKI is uniformly preferred to any system, as is appropriate for an upper bound. The S2S model is the least-preferred with respect to WIKI. The S2S+AE model is more-preferred than the BASE and S2S models, by a larger margin for the latter. These results show that without autoencoding, the sequence-to-sequence model is less effective than a template-based system. Finally, although WIKI is more preferred than S2S+AE, the distributions are not significantly different, which we interpret as evidence that the model is able to generate good text from the human point-of-view, but autoencoding is required to do so.

7 Analysis

While results presented above are encouraging and suggest that the model is performing well, they are not diagnostic in the sense that they can drive deeper insights into model strengths and weaknesses. While inspection and manual analysis is still required, we also leverage the structured factual data inherent to our task to perform quantitative as well as qualitative analysis.

7.1 Fact Count

Figure 4 shows the effects of input fact count on generation performance. While more input facts give more information for the model to work with, longer inputs are also both rarer and more complex to encode. Interestingly, we observe the S2S+AE model maintains performance for more complex inputs while S2S performance declines.

7.2 Example generated text

Table 5 shows some DEV entities and their summaries. The model learns interesting mappings: between numeric and string dates, and country de-

Data		COUNTRY_OF_CITIZENSHIP united states of america DATE_OF_BIRTH 16/04/1927 DATE_OF_DEATH 19/05/1959 OCCUPATION formula one driver PLACE_OF_BIRTH redlands PLACE_OF_DEATH indianapolis SEX_OR_GENDER male TITLE bob cortner
WIKI	n/a	**robert** *charles* **cortner** (**april 16 , 1927 may 19 , 1959**) was an **american automobile racing driver** from *redlands , california* .
BASE	47.7	**bob cortner** (born **16 april 1927** in **redlands** ; died **19 may 1959** in **indianapolis**) was a **formula one driver** from the **united states of america**
S2S	45.7	**bob cortner** (**april 16 , 1927** ~~may 19 , 2005~~) was an **american** ~~professional boxer~~ .
S2S+AE	58.8	**robert cortner** (**april 16 , 1927 may 19 , 1959**) was an **american race-car driver** .
Data		COUNTRY_OF_CITIZENSHIP united kingdom DATE_OF_BIRTH 08/01/1906 DATE_OF_DEATH 12/12/1985 OCCUPATION actor PLACE_OF_BIRTH london PLACE_OF_DEATH chelsea SEX_OR_GENDER male TITLE barry mackay (actor)
WIKI	n/a	**barry mackay** (**8 january 1906 12 december 1985**) was a **british actor**.
BASE	34.3	**barry mackay (actor)** (born **8 january 1906** in **london** ; died **12 december 1985 in chelsea**) was an **actor** from the **united kingdom** .
S2S	84.8	**barry mackay** (**8 january 1906 12 december 1985**) was a **british** ~~film actor~~ .
S2S+AE	76.7	**barry mackay** (**8 january 1906 12 december 1985**) was an **english actor** .
Data		COUNTRY_OF_CITIZENSHIP united states of america DATE_OF_BIRTH 27/08/1931 DATE_OF_DEATH 03/11/1995 OCCUPATION jazz musician SEX_OR_GENDER male TITLE joseph ”flip” nuñez
WIKI	n/a	**joseph** *“ flip ’* **nuñez** was an **american** ~~jazz pianist~~ , ~~composer~~ , and ~~vocalist~~ of *filipino* descent .
BASE	15.0	**joseph** *“ flip ’* **nuñez** (born **27 august 1931** ; died **3 november 1995**) was a **jazz musician** from the **united states of america** .
S2S	29.1	**joseph** *“ flip ’* **nuñez** (**august 27 , 1931 november 3 , 1995**) was an **american** ~~jazz trumpeter~~ .
S2S+AE	29.1	**joseph** *“ flip ’* **nuñez** (**august 27 , 1931 november 3 , 1995**) was an **american** ~~jazz drummer~~ .

Table 5: Examples of entities from DEV, showing facts, WIKI, BASE, S2S and S2S+AE. We mark **correct**, ~~incorrect~~ and *extra* fact values in the text with respect to the Wikidata input.

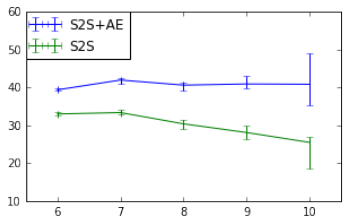

Figure 4: BLEU vs Fact Count on instances from DEV. Error bars indicate the 95% confidence interval for BLEU.

monyms. The model also demonstrates the ability to work around edge cases where templates fail, i.e. stripping parenthetical disambiguations (e.g. (actor)) and emitting the name Robert when the input is Bob. Output also suggests the model may perform inference across multiple facts to improve generation precision, e.g. describing an entity as english rather than british given information about both citizenship and place of birth. Unfortunately, the model can also infer unsubstantiated facts into the text (i.e. jazz drummer).

7.3 Content selection and hallucination

We randomly sample 50 entities from DEV and manually annotate the Wikipedia and system text. We note which fact slots are expressed as well as whether the expressed values are correct with respect to Wikidata. Given two sets of correctly extracted facts, we can consider one *gold*, one *system* and calculate set-based precision, recall and F1.

What percentage of facts are used in the reference summaries? Firstly, to understand how Wikipedia editors select content for the first sentence of articles, we measure recall with the real facts as gold, and Wikipedia as system. Overall, the recall is 0.61 indicating that 61% of input facts are expressed in the reference summary from Wikipedia. The entity name (TITLE) is always expressed. Four slots are nearly always expressed when available: OCCUPATION (90%), DATE_OF_BIRTH (84%), CITIZENSHIP (81%), DATE_OF_DEATH (80%). Six slots are infrequently expressed in the analysis sample: PLACE_OF_BIRTH (33%), POSITION_HELD (25%), PARTICIPANT_OF (20%), POLITICAL_PARTY (20%), EDUCATED_AT (14%), SPORTS_TEAM (9%). Two are never expressed explicitly: PLACE_OF_DEATH (0%), SEX_OR_GENDER (0%). AWARD_RECEIVED and SPORT are not in the analysis sample.

Do systems select the same facts found in the reference summaries? Table 6 shows content selection scores for systems with respect to the Wikipedia text as reference. This suggests that the autoencoding in S2S+AE helps increase fact recall without sacrificing precision. The template baseline also attains this higher recall, but at the cost of precision. For commonly expressed facts found in most person biographies, recall is over 0.95 (e.g., CITIZENSHIP, BIRTH_DATE, DEATH_DATE and OCCUPATION). Facts that are infrequently expressed are more difficult to select, with system F1 ranging from 0.00 to 0.50. Interestingly, macro-averaged F1 across infrequently expressed facts mirror human preference rather than BLEU results, with S2S+AE (0.26) > BASE (0.17) > S2S (0.07). However, all systems perform poorly on these facts and no reliable differences are observed.

How does autoencoding effect fact density? Interestingly, we observe that the autoencoding objective encourages the model to select more facts (5.2 for S2S+AE vs. 4.5 for S2S), without increasing sentence length (19.1 vs. 19.7 tokens). BASE is similarly productive (5.1 facts) but wordier (21.2 tokens), while the WIKI reference produces both more facts (6.1) and longer sentences (23.7).

	P	R	F
BASE	0.80	0.79	0.79
S2S	0.89	0.67	0.77
S2S+AE	0.89	0.78	0.83

Table 6: Fact-set content selection results phrased as precision, recall and F1 of systems with respect to the Wikipedia reference on DEV.

	P	R	F
BASE	1.00	0.74	0.85
S2S	0.96	0.55	0.70
S2S+AE	0.93	0.62	0.74
WIKI	0.81	0.61	0.69

Table 7: Hallucination results phrased as precision, recall and F1 of systems with respect to the Wikidata input on DEV.

Do systems hallucinate facts? To quantify the effect of hallucinated facts, we asses content selection scores of systems with respect to the input Wikidata relations (Table 7). Our best model achieves a precision of 0.93 with respect to Wikidata input. Notably, the template-driven baseline maintains a precision of 1.0 as it is constrained to emit Wikidata facts verbatim.

8 Discussion and future work

Our experiments show that RNNs can generate biographic summaries from structured data, and that a secondary autoencoding objective is able to account for some of the information mismatch between input facts and target output sentences. In the future, we will explore whether results improve with explicit modelling of facts and conditioning of generation and autoencoding losses on slots. We expect this could benefit generation for diverse and noisy slot schemas like Wikipedia Infoboxes.

Another natural extension is to investigate the performance of the network running in reverse, from summary text back to facts. We plan to isolate the performance of the S2S+AE backward model when inferring facts and compare it to stan-

dard relation extraction systems. Finally, similar RNN models have been applied extensively to language translation tasks. We plan to explore whether a joint model of machine translation and fact-driven generation can help populate KB entries for low-coverage languages by leveraging a shared set of facts.

9 Conclusion

We present a neural model for mapping between structured and unstructured data, focusing on creating Wikipedia biographic summary sentences from Wikidata slot-value pairs. We introduce a sequence-to-sequence autoencoding RNN which improves upon base models by jointly learning to generate text and reconstruct facts. Our analysis of the task suggests evaluation in this domain is challenging. In place of a single score, we analyse statistical measures, human preference judgements and manual annotation to help characterise the task and understand system performance. In the human preference evaluation, our best model outperforms template baselines and is preferred 40% of the time to the gold standard Wikipedia reference.

Code and data is available at `https://github.com/andychisholm/mimo`.

Acknowledgments

This work was supported by a Google Faculty Research Award (Chisholm) and an Australian Research Council Discovery Early Career Researcher Award (DE120102900, Hachey). Many thanks to reviewers for insightful comments and suggestions, and to Glen Pink, Kellie Webster, Art Harol and Bo Han for feedback at various stages.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Conference on Empirical Methods in Natural Language Processing*, pages 502–512.

David Bamman and Noah A. Smith. 2014. Unsupervised discovery of biographical structure from text. *Transactions of the Association for Computational Linguistics*, 2:363–376.

Fadi Biadsy, Julia Hirschberg, and Elena Filatova. 2008. An unsupervised approach to biography production using Wikipedia. In *Annual Meeting of the Association for Computational Linguistics*, pages 807–815.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media.

Sasha Blair-Goldensohn, David Evans, Vasileios Hatzivassiloglou, Kathleen McKeown, Ani Nenkova, Rebecca Passonneau, Barry Schiffman, Andrew Schlaikjer, Advaith Siddharthan, and Sergey Siegelman. 2004. Columbia University at DUC 2004. In *Proceedings of the Document Understanding Workshop*, pages 23–30.

Gong Cheng, Danyun Xu, and Yuzhong Qu. 2015. Summarizing entity descriptions for effective and efficient human-centered entity linking. In *International Conference on World Wide Web*, pages 184–194.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Annual Conference on Neural Information Processing Systems*, pages 3079–3087.

Pablo Ariel Duboue and Kathleen R McKeown. 2003. Statistical acquisition of content selection rules for natural language generation. In *Conference on Empirical Methods in Natural Language Processing*, pages 121–128.

Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. In *International Conference on Computational Semantics*, pages 83–94.

Nikesh Garera and David Yarowsky. 2009. Structural, transitive and latent models for biographic fact extraction. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 300–308.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1296–1306.

Bikash Gyawali and Claire Gardent. 2014. Surface realisation from knowledge-bases. In *Annual Meeting of the Association for Computational Linguistics*, pages 424–434.

Sangdo Han, Jeesoo Bang, Seonghan Ryu, and Gary Geunbae Lee. 2015. Exploiting knowledge base to generate responses for natural language dialog listening agents. In *Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 129–133.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Workshop on Statistical Machine Translation*, pages 187–197.

Ravi Kondadadi, Blake Howald, and Frank Schilder. 2013. A statistical NLG framework for aggregated planning and realization. In *Annual Meeting of the*

Association for Computational Linguistics, pages 1406–1415.

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

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 71–78.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *International Conference on Learning Representations*.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2015. What to talk about and how? Selective generation using LSTMs with coarse-to-fine alignment. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 720–730.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 145–152.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Richard Power and Allan Third. 2010. Expressing OWL axioms by english sentences: Dubious in theory, feasible in practice. In *International Conference on Computational Linguistics*, pages 1006–1013.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Barry Schiffman, Inderjeet Mani, and Kristian Concepcion. 2001. Producing biographical summaries: Combining linguistic knowledge with corpus statistics. In *Annual Meeting of the Association for Computational Linguistics*, pages 458–465.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Annual Conference on Neural Information Processing Systems*, pages 3104–3112.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In *ICML Deep Learning Workshop*.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Annual Conference on Neural Information Processing Systems*, pages 2755–2763.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.

Chunyang Xiao, Marc Dymetman, and Claire Gardent. 2016. Sequence-based structured prediction for semantic parsing. In *Annual Meeting of the Association for Computational Linguistics*, pages 1341–1350.

Lanbo Zhang, Yi Zhang, and Yunfei Chen. 2012. Summarizing highly structured documents for effective search interaction. In *International Conference on Research and Development in Information Retrieval*, pages 145–154.