Domain Transfer for Deep Natural Language Generation from Abstract Meaning Representations

Nina Dethlefs
School of Engineering and Computer Science
University of Hull, UK

Abstract

Stochastic natural language generation systems that are trained from labelled datasets are often domain-specific in their annotation and in their mapping from semantic input representations to lexical-syntactic outputs. As a result, learnt models fail to generalize across domains, heavily restricting their usability beyond single applications. In this article, we focus on the problem of domain adaptation for natural language generation. We show how linguistic knowledge from a source domain, for which labelled data is available, can be adapted to a target domain by reusing training data across domains. As a key to this, we propose to employ abstract meaning representations as a common semantic representation across domains. We model natural language generation as a long short-term memory recurrent neural network encoder-decoder, in which one recurrent neural network learns a latent representation of a semantic input, and a second recurrent neural network learns to decode it to a sequence of words. We show that the learnt representations can be transferred across domains and can be leveraged effectively to improve training on new unseen domains. Experiments in three different domains and with six datasets demonstrate that the lexical-syntactic constructions learnt in one domain can be transferred to new domains and achieve up to 75-100% of the performance of in-domain training. This is based on objective metrics such as BLEU and semantic error rate and a subjective human rating study. Training a policy from prior knowledge from a different domain is consistently better than pure in-domain training by up to 10%.

I. Introduction

NATURAL language generation (NLG) is the task of finding a natural language description for a non-linguistic input representation, such as a database entry, dialogue act or other non-linguistic object. Traditionally, NLG has been seen to consist of three stages: (a) text planning, which involves content determination and discourse structuring, (b) microplanning, which involves lexicalization, referring expression generation and aggregation, and (c) realization, which involves linguistic and structure realization [1]. In this article, we focus on the latter two tasks. We assume that a semantic input form is provided by a pre-processing module, such as a dialogue manager or route planner, and that the content to be conveyed has already been determined. We are thus particularly concerned with the task of expressing a given semantic input form as a syntactically well-formed sequence of words.

Common approaches to this problem have been rule-based, see e.g. [1], grammar-based, e.g. using combinatory categorical grammar (CCG) [2], or based on supervised learning, e.g. [3]–[7], or reinforcement learning [8], [9]. A common issue to all of these approaches is that a significant amount of human effort is required to adapt a language generator to a new domain. This effort can be spent on the design of rules, grammar engineering, or in the case of stochastic language generators, on the collection and annotation of data. Once trained, stochastic language generators that are trained from annotated corpora are typically poor at generating outputs for unseen semantic inputs or for new domains. A common problem is the incompatibility of input representations across domains as inputs are often specific to the non-linguistic data structures that are conventionally used in the target domain, e.g. route instructions, weather, or spoken dialogue applications. Approaches that have trained models for multiple domains typically rely on specialized resources such as annotated databases [10], require substantial linguistic pre-processing to facilitate grammar extension [11], or work only when the input representation across

Corresponding author: Nina Dethlefs (Email: n.dethlefs@hull.ac.uk)

domains is identical [12]. The lack of generalizability across (even similar) inputs severely limits the reuse of language generators beyond their original domain.

Other areas of natural language processing (NLP) have made increasing use of deep learning and reported remarkable results. Examples include parsing [13], machine translation [14] sentiment analysis [15], language understanding [16], [17] and speech recognition [18], to name just a few. A benefit of deep learning models is their ability to discover common patterns, e.g. phrase structure, across a large number of examples—which can also be useful for language generation.

To circumvent the problem of incompatible input representations, we propose in this article that using abstract meaning representations (AMRs) as a common representation language across domains has the advantage that data from one domain can be reused in another domain without problems. While the occurrence statistics of lexical-syntactic patterns will vary across domains, e.g. route instructions will have a high proportion of prepositional phrases and spatial relations, and referring expressions will have a high proportion of noun phrases, the general learnt mappings from semantic inputs to lexical-syntactic outputs remain transferable across tasks. In this way, our multi-task learning problem can be seen as a domain adaptation problem, and we can—similarly to other areas of NLP—make use of deep learning to learn common patterns across domains.

To do this, we model our natural language generator as a Long Short-Term Memory (LSTM) encoder-decoder, in which two LSTMs are jointly trained to learn a probability distribution that conditions a sequence of words on a sequence of semantic symbols. One LSTM (the encoder) learns a hidden representation of a sequence of semantic inputs. A second LSTM (the decoder) learns to decode this hidden representation to a sequence of words, which represents a surface realization of the semantic input. The encoder-decoder model was proposed by Cho et al. [19], who applied it to Statistical Machine Translation (SMT) and is related to the sequence-to-sequence classification model by Sutskever et al. [20], who also worked on SMT.

We evaluate our model in three domains that are representative of different areas of interest in NLG: referring expression generation, route instruction generation and spoken dialogue applications. We compare the following training scenarios: (1) in-domain training of our encoder-decoder model, where a language generator is trained and tested in the same domain, (2) out-of-domain training, where a model is trained in a source domain but tested in a target domain, and (3) training an in-domain policy from out-of-domain prior knowledge. Results show that the model developed for the third scenario performs best according to both objective and subjective measures. A qualitative analysis shows that our model is able to learn abstract patterns of basic linguistic constructions, such as transitive and intransitive sentence patterns, spatial relations, relative clauses, complex noun phrases and basic discourse relations such as ‘and’ or ‘but’. All code and data for this article are available.[1]

The remainder of this article is structured as follows. Section II discussed related work in the areas of deep learning, NLG and domain adaptation. Section III introduces the main learning model for our research. Section IV describes the datasets and abstract meaning representations used for training of our models. Section V describes the training and evaluation scenario and discusses the main objective and subjective results. Section VI, finally, draws conclusions and discusses possibilities for future work.

II. Related Work

A. Stochastic Corpus-based Natural Language Generation

Stochastic natural language generators that find a lexical-syntactic realization for a semantic input can be time-consuming and expensive to build for new domains, especially when data needs to be collected and labelled from scratch. Consequently, recent years have seen an increased interest in approaches that can induce a natural language generator (semi-) automatically either from raw data or from parallel or aligned datasets. Learning from raw data typically requires finding an alignment between phrases in a

[1]http://www.hull.ac.uk/php/496827/ieeeci-materials.zip

text that mean the same thing. For example, Liang et al. [21] present an approach that finds an alignment between a set of records (e.g. weather events or moves in a RoboCup game) with a textual description of the event. Similarly, Cuayáhuitl et al. [22] trained a semantic slot labeller, which automatically annotates raw text to be used for training of a language generator. Automatic annotations are prepared based on alignments of syntactic phrases in the input data that have a common label. Both approaches reported good results but will in practice depend on the quality and correctness of the alignment, and thus presumably on the linguistic diversity and complexity of the dataset.

Other approaches have instead taken advantage of the existence of parallel datasets for some domains, such as annotated databases [6], [23]. NLG in these cases boils down to learning an accurate mapping between combinations of database entries and possible surface forms. Both types of approaches have shown promising performance in the past for their respective domains, but are heavily reliant on the existence (or collection) of parallel resources.

Another alternative to aligned data is the work by Chen and Mooney [24] and MacMahon et al. [25], who learnt an automatic mapping between a set of natural language instructions and a set of action sequences that carry them out. More recent work by Mei et al. [26] has revisited this problem and shown that deep learning—an LSTM encoder-decoder as we also use—can outperform earlier approaches on the same task. Daniele et al. [27] presented work on NLG for navigation instruction generation using a combination of inverse reinforcement learning (for content selection) and surface realization using a sequence-to-sequence LSTM. Oswald et al. [28] similarly used inverse reinforcement learning to generate instructions from a corpus of human examples.

B. Deep Learning for Natural Language Generation

Previous work on deep learning for NLG has taken one of two routes, either exploiting the potential bidirectionality of models or treating generation as a sequence prediction problem.

The first strand of work typically assumes that a latent representation of the input can be learnt during an encoding stage, e.g. using an autoencoder, so that the original inputs can be reconstructed in a decoding stage. The latter can then be exploited for language generation. As an example of this approach, Iyyer et al. [29] applied the Recursive Autoencoder model in Socher et al. [13] to dependency parse trees and reconstruct previously encoded inputs as a way to do paraphrase generation. The model successfully re-generates short input sequences of 2-3 words. Similar results are reported in Andreas and Ghahramani [30] and in Dinu and Baroni [31], where bidirectional models transform language into a distributional semantic representation and then back into language.

The second and alternative strand of work has treated NLG as a sequence prediction task in which the next symbol in a sequence depends on the context of preceding symbols. This technique was first introduced by Mikolov et al. [32], who applied a recurrent neural network (RNN) architecture to predict sequences of words in order to obtain better speech recognition results. Sutskever et al. [33] applied a similar model to predict sequences of characters to form strings of words. In an application of NLG to spoken dialogue, Wen et al. [34] used an LSTM for sentence planning and surface realization. The authors treated generation as a next-word-in-sequence prediction task. Input to the model is a dialogue act with delexicalized slot values. Dusek and Jurcicek [35] showed how a sequence-to-sequence model for NLG can generate outputs from unaligned training data and outperform previous work [5] that relied on aligned semantic inputs and lexical-syntactic outputs. The generator proposed by Dusek and Jurcicek [35] operates in two alternative modes, either producing natural language strings directly from dialogue acts, or generating syntactic dependency trees as an intermediate step. Our setting lies perhaps in the middle, in that we generate from linguistic AMR representations but do not use dialogue acts as inputs to the generation process.

Our learning model is closely related to the encoder-decoder model of Cho et al. [19] and the sequence-to-sequence classification approach of Sutskever et al. [20], both of which were first applied to statistical machine translation. The basic idea is to learn an encoding of a sequence in a source language using one

RNN and then learn a decoding to a sequence in the target language using a separate RNN. By training both models jointly, a mapping from the source to the target language can be learnt. The approach in Sutskever et al. [20] is based on an LSTM model, while Cho et al. [19] used a Gated Recurrent Unit (GRU). Chung et al. [36] made a comparison of both and showed that they yield similar performance in practice.

In a wider context, our work is also related to work on summarization that uses the LSTM encoder-decoder model, such as Chopra et al. [37] or Gerani et al. [38], where the focus is on abstractive sentence summarization, and Mei et al. [39], where the authors generated language by jointly optimising content selection and surface realization. The latter two approaches also made use of an attention mechanism.

C. Domain Adaptation and Multi-task Learning

The idea of reusing knowledge from a source domain in a new target domain is not new to other areas of NLP, such as part-of-speech (POS) tagging, named entity recognition, capitalization or shallow parsing. The main principle is to identify which features in the data are suitable for sharing and which are not. As an example of reusing prior knowledge, Chelba and Acero [40] used probabilities learnt in a source domain as prior probabilities in a target domain, and subsequently adapted them to their new context during training. In an alternative approach, Daumé-III [41] proposed the use of an augmented feature set, including features specific to the source domain, specific to the target domain, and shared between both. The knowledge reuse can then focus on the shared feature set.

A recent approach to multi-domain NLG was presented by Wen et al. [12]. The authors generate synthetic training data to adapt resources from a source domain to a target domain, but restrict themselves to semantic slots that exist in both domains. Also, both source and target domain, TVs and laptops, were chosen to be relatively close (semantically) to ease the domain transfer. Given the specificity of input representations in Wen et al. [12], it is not clear that the model is general enough to learn basic linguistic patterns that can be transferred across semantically more distant domains as was shown for work in other areas of NLP.

Recent approaches to domain adaptation in a variety of NLP tasks have shown that combining data (with a common input representation) can lead to significant improvements in individual domains [40]–[42]. To address the problem that some NLP tasks, such as natural language understanding or generation, often rely on domain-specific representations, Kim et al. [43] presented an approach that clusters labels to find possible mappings. The authors use canonical correlation analysis to identify distinct labels that occur in similar contexts across domains and can thus be assigned the same pseudo-label to assist domain transfer. Results report better performance on a joint domain model than a single domain model. This is confirmed in work by Jaech et al. [44], where the authors showed that training a multi-task model for slot filling in language understanding from several domains gives significantly better performance than training models for single domains. In their model, the authors use a bidirectional LSTM which remains general across tasks with the only domain specificity lying in the embedded semantics—which are trained per domain. In an approach to dialogue management for multiple domains, Cuayáhuitl et al. [45], [46] use deep reinforcement learning to extract behaviours from a meta-domain that is transferable to multiple specific domains, and show that domain transfer is possible from unlabelled input examples.

We present a novel approach to domain adaptation for NLG that represents data across domains in a common input representation. In this way, we are able to learn basic linguistic patterns from multiple domains and reproduce the positive effects of multi-domain training reported for other areas of NLP.

III. Learning Model

A. Recurrent Neural Networks

An RNN is a type of neural network that learns a hidden representation $\mathbf{h}$ of an input sequence $\mathbf{x} = (x_1, \ldots, x_N)$ by learning an increasingly abstract encoding of the inputs. An RNN can also have an

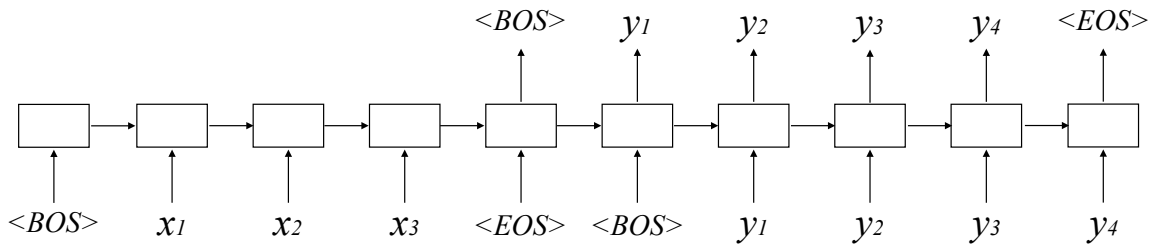

Fig. 1. Illustration of the sequence-to-sequence learning model. An input sequence $\langle BOS\rangle, x_1, x_2, x_3, \langle EOS\rangle$ is encoded, where $\langle BOS\rangle$ refers to the beginning-of-sequence symbol and $\langle EOS\rangle$ refers to the end-of-sequence symbol. The learnt hidden representation is then decoded to sequence $\langle BOS\rangle, y_1, y_2, y_3, y_4, \langle EOS\rangle$. In our case, the input sequence corresponds to a sequence of semantic symbols and the output sequence is a sequence of words.

output sequence $\mathbf{y} = (y_1, \ldots, y_M)$, which can be reconstructed from $\mathbf{h}$. We assume that $\mathbf{x}$ and $\mathbf{y}$ can have different lengths. The hidden representation $\mathbf{h}$ can be found through updates at time step t:

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, x_t), \tag{1}$$

where f is an activation function, such as a sigmoid, tangent or ReLU. During training, the goal is to minimise the loss L between the input and desired output of the RNN:

$$L(x, y) = -\frac{1}{N}\sum_{n \in N} x_n \log y_n, \tag{2}$$

for which we will use cross entropy. In the model we present, the input sequence $\mathbf{x}$ will correspond to a sequence of semantic symbols, for which we wish to find a natural language expression, for example: `(b / ball :domain (n / entity) :mod red)`. The output sequence $\mathbf{y}$ will correspond to a sequence of words expressing $\mathbf{x}$, for example: "the red ball" or "the ball that is red", where the exact realization remains underspecified in the semantic input. Both sequences are represented as 1-hot vectors, where each vector contains a single 1 to represent the occurrence of a semantic symbol (for input sequences) or a word (for output sequences). The goal is to learn a probability distribution that conditions a target sequence on a source sequence. All input and output sequences start with the special symbol `BOS` and end in `EOS` to mark the beginning and end of sequence, respectively. An illustration of our model is shown in Figure 1.

B. The LSTM Encoder-Decoder

As conventional update functions, such as sigmoid or tangent, have been associated with the problem of vanishing or exploding gradients [47], we use an LSTM [48] to implement our encoder-decoder. In contrast to a conventional RNN, an LSTM has three gates, which control the loss and addition of information for the current "cell state". Each gate has the same shape as the hidden state. The "input gate" i is a sigmoid function which determines how much new available information to add to the cell state at the current time step. It first identifies for each member of the cell state vector whether it should be updated or not, and then chooses an update from a set of candidates. The "forget gate" f is a sigmoid function that determines for each member in the cell state vector whether it should be forgotten or retained. Finally, the "output gate" o determines what the output of the cell state should be.

Both the encoder and decoder LSTM follow the definition by Graves [49], according to which we update the hidden state h at each time step t using the following steps:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + bi) \tag{3}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + bf) \tag{4}$$

TABLE I

SUMMARY OF DATASETS USED FOR TRAINING AND THEIR COVERAGE OF LINGUISTIC PATTERNS. LINGUISTIC FEATURES SHOWN ARE AS FOLLOWS: NUMBER OF EXAMPLES (EXAMPLES), VOCABULARY SIZE (VOCAB. SIZE), AVERAGE EXAMPLE LENGTH (AVE. LEN.), NUMBER OF NPS (NPS), NUMBER OF SPATIAL RELATIONS (SRS), NUMBER OF TRANSITIVE CLAUSES (TRANS. CL.), NUMBER OF INTRANSITIVE CLAUSES (INTR. CL.), NUMBER OF RELATIVE CLAUSES (REL. CL.), NUMBER OF IMPERATIVES (IMPERATIVES) AND NUMBER OF CONJUNCTIONS (CONJ.).

Dataset	Examples	Vocab. size	Ave. len.	NPs	SRs	Trans. Cl.	Intr. Cl.	Rel. Cl.	Imperatives	Conj.
GRE	4,480	195	3.50	4,969	1,084	45	0	15	0	3
REFCOCO	142,051	10,046	3.50	302,703	54,484	384	6,634	794	229	2,329
GIVE	1,756	467	3.30	706	1,716	294	17	4	1,043	100
SAIL	816	295	7.95	598	895	218	64	5	656	135
SFXR	6,198	2,058	12.91	6,094	956	2,597	872	598	3	905
SFXH	6,384	1,061	12.13	6,403	1,192	2,247	938	761	1	850

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc} x_t + W_{hc} h_{t-1} + bc) \tag{5}$$

$$o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + W_{co} c_t + bo) \tag{6}$$

$$h_t = o_t \tanh(c_t) \tag{7}$$

In Equations (3)-(7), σ refers to the logistic sigmoid function, and i, f, o and c refer to the input gate, forget gate, output gate and cell state vectors, respectively. We use a deep LSTM with 4 layers and follow Sutskever et al. [20] in inverting the symbols in the input sequence $\mathbf{x}$, which was shown to lead to better performance in previous work, presumably due to the short-term dependencies it creates between input and output symbols.

IV. DOMAIN TRANSFER FOR NATURAL LANGUAGE GENERATION

A. Domains and Data

We will apply our model to three different domains which have traditionally been of interest to NLG: referring expressions, spatial navigation and spoken dialogue. Each domain is represented by two separate datasets, one serving as a source and the other serving as a target domain. All six datasets are summarized in Table I in terms of their size and coverage of linguistic patterns.

In terms of referring expressions, GRE [50] contains referring expressions in a 3D scene. The focus is on noun phrases and spatial relations. GRE will serve as a source domain for referring expressions. REFCOCO contains identifying descriptions of people and objects in images [51]. It will serve as a target domain.

The GIVE corpus represents our source domain in spatial navigation. It contains the set of 63 English dialogues collected by Gargett et al. [52] in the GIVE task, *Generating Instructions in Virtual Environments*. GIVE involves two participants that engage in a "treasure hunt", where one participant instructs another so as to navigate through a virtual world in order to find a trophy. Instructions are of a spatial and navigation nature, mixed with referring expressions to buttons that need to be pressed. We use the SAIL corpus as a target domain [24], [25], which also contains navigation instructions across a virtual grid environment.

For spoken dialogue, we use the SFX-restaurants (SFXR) and SFX-hotels (SFXH) corpora, which were collected by Wen et al. [34]. Both datasets contain utterances that a spoken dialogue system might produce in prompting the user for their search queries, presenting results or confirming slots.

Each dataset contains different distributions of linguistic patterns. The GRE corpus, e.g. contains a high number of simple and complex noun phrases, such as "green sphere" and "red ball leaning against the red cube". The GIVE corpus includes a high number of spatial relations and prepositional phrases as well as imperatives, e.g. "turn around and exit the room" and "turn left 90 degrees". The SFXR and SFXH datasets contain the highest number of transitive constructions, relative clauses and questions: "Ar Roi Restaurant serves Thai food.", "What price range are you looking for?" or "B Star Bar is a moderately priced

Alternative sample realisations:

1a. Beijing restaurant is serving Chinese food.
1b. Beijing restaurant is a nice place. It serves Chinese food.
1c. Beijing restaurant is a Chinese restaurant.

Flat semantic representation (for realizations 1a.-1c.)

```
inform(name='beijing_restaurant', food=chinese)
```

AMR and corresponding tree representation (for realization 1a.)

```
(s / serve
     :arg0 (n / restaurant
              :name 'beijing_restaurant')
     :arg1 (f / food :mod chinese)
)
```

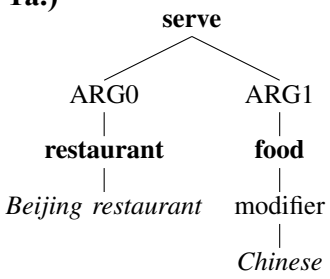

Fig. 2. Example involving three alternative descriptions of a restaurant, realizations 1a.-1c. While the example of a flat semantic input representation is the same for all three realizations, an AMR can capture semantic and syntactic differences at a finer level of granularity.

restaurant that serves Asian food." While our target datasets represent similar patterns in the respective domains, they occur with different distributions. For example, REFCOCO contains a substantially higher number of spatial relations than GRE, and SAIL contains fewer referring expressions of the kind found in GIVE.

B. Abstract Meaning Representations

To adapt linguistic knowledge learnt in one domain to another, it is important that the semantic input representations are compatible across datasets. We therefore represent all semantic forms as AMRs [53]. They are acyclic directed graphs that draw on a set of common semantic categories and abstract away from syntactic peculiarities. They have recently been adopted for parsing [54], CCG semantic parsing [55], summarization [56], and NLG [57] with the aim of adopting a more standardized representation across tasks, models and frameworks. AMRs offer a number of advantages over flat sequential structures that are frequently used as input and often correspond to specific semantic slots that need to be expressed.

Consider the flat semantic structure in Figure 2 and its three sample realizations 1a-1c. Transferring such underspecified representations across domains is difficult, even if we adopted the same label names across domains, because it is not clear which surface realization can be transferred and which cannot. Also, it is often not possible to account for semantic nuances with flat structures. In example 1b., the surface realization contains the modifier "nice", which is not in the semantic form. As flat structures tend to cover an exact set of semantic slots, additional information, such as a restaurant being "nice", can often get lost. Alternatively, information can get inserted in surface forms when it is not intended. The word "nice" might be learnt as part of the whole construction and then inserted in surface forms when the restaurant in question is not actually "nice". Training from a corpus in which not all information is annotated can therefore lead to semantically inaccurate outputs, and while this problem can of course be mitigated by collecting corpora for each new domain, it precludes the use of existing natural datasets.

Expressing all semantics as AMRs allows us to make more fine-grained semantic distinctions for alternative realizations while still leaving details such as number, tense, voice or discourse relations underspecified. Similar findings were made for other linguistically-informed frameworks, e.g. in Chenar et al. [58]. The authors introduced an attention model for LSTMs that is guided by linguistic knowledge, and can in this way identify the most salient linguistic structures in a dataset. Their results focus on language understanding, and show how linguistic knowledge can be leveraged to improve the generalizability of deep learning models trained from small datasets.

The idea of a linguistically-informed NLG process is not new. Earlier work on systemic functional grammar has treated NLG as a network of consecutive choices that ultimately lead to a semantically, socially and textually appropriate output [59]. The idea of using an abstract semantic representation to generalize across more specific syntactic realizations and even different languages has particularly been advocated in Meaning-Text Theory (MTT) [60]. MTT postulates an independent semantic level that allows the specification of abstract meanings that are independent of sound and structure. These meanings are represented as graphs (or trees) and lay out the main relationships and dependencies between entities and events. Each meaning representation can map to a potentially large number of surface realizations, and language-dependent knowledge is kept in a rich lexicon [61]. MTT has been a popular choice in multi-lingual NLG [62], [63] for its feature of allowing the separation of language-specific and language-independent resources.

Recent efforts in linguistically-informed NLG include SimpleNLG [64], a toolkit for the development of practical NLG applications. SimpleNLG similarly allows the separation of general and domain-specific considerations. The tool provides operations for combining major syntactic categories in English, while lexical specificities can be dealt with in particular application domains, e.g. the grammatical case system for some languages. Varieties of SimpleNLG also exist for other languages such as German [65], French [66] or Italian [67]. SimpleNLG could probably cover many aspects of our target domains, if our focus was not on domain transferability of data-driven resources.

For this article, we prepared AMR annotations for all six datasets above. The annotations were prepared semi-automatically and corrected by hand by two researchers. This process took us about 2 hours per 100 sentences. The semi-automatic process involved finding identical utterances in the datasets and making sure that they receive the same AMR. This was easier in the (smaller) GRE and GIVE datasets than in the larger ones. For REFCOCO, which is much larger than the other datasets, we used the Stanford POS tagger [68] on all utterances and then assigned the same AMR structure for utterances with the same POS sequence. We followed the same annotation procedure for the target domains.

C. Reusing Training Data Across Domains

An obvious drawback of the AMR inputs is that flat structures are presumably much easier to obtain automatically from a database or other non-linguistic representation of the domain. We expect, however, that using AMRs, we can learn enough linguistic regularities to require very few or no additional annotations for a new domain. We will test this hypothesis in Section V-A.

To train an RNN natural language generator as described in Section III that generalizes across all our domains, we follow two steps: We replace all domain-specific concepts by a more general semantic category. Specifically, we refer to all concepts that correspond to nouns, such as “restaurant”, “button”, “food”, etc. as *object*, to all verb-corresponding concepts, such as “serve”, “recommend”, or “find” as *event*, and to all adjectives and modifiers as *property*. This will ensure that AMR structures are indeed abstract and e.g. the AMR for a transitive construction will indeed generalize to new domains regardless of the specific concepts involved.

V. TRAINING AND EVALUATION

A. Learning Setup and Baselines

We evaluate our model in three settings:

1) An LSTM encoder-decoder model with **in-domain** training. A model is trained from 80% of annotated training data in the target domain and evaluated based on the remaining 20%.
2) The same model with **out-of-domain** training. A model is trained in a source domain and the learnt weights are then evaluated in a new target domain.
3) The same model with out-of-domain **prior**. Here, we use the out-of-domain model trained from a source domain and use it as a prior to in-domain training from target data.

All models were trained with mini-batch gradient descent with Adam optimization and a batch size of 128. We trained each model for 10,000 epochs which took between 3 and 6 days—depending on the complexity and number of training examples—on a Tesla K40 GPU and a Titan X Pascal GPU. All code was implemented in Keras [69] using a Theano backend [70]. The size of input/output sequences $\mathbf{x}$ and $\mathbf{y}$ was the sum of unique vocabulary items and semantic symbols in each domain (after delexicalization) and we use 50 hidden nodes.

B. Results: Objective Evaluation

We evaluate our models using the BLEU metric [71] and compare the highest-ranked output candidate for a semantic input against its human references from the test set. We report BLEU scores for 3-grams and 4-grams. We also report the semantic error in generated outputs, which is computed as $\text{ERR} = \frac{a+b}{C}$, where a is the number of slots missed in the realization, b is the number of superfluous slots, and C is the total number of slots.

Objective results are shown in Table II, which contrasts results in all three evaluation scenarios against a human upper-bound. We train models for each of our source domains GRE, GIVE and SFXR and evaluate them in target domains REFCOCO, SAIL and SFXH. We also present the in-domain performance for each of the six datasets as a comparison. As an additional comparison, we present results that test GIVE with GRE as a source domain and GRE with GIVE as a source domain. The latter is interesting because GIVE contains similar referring expressions to GRE.

We can see that *in-domain* training performs well in all domains achieving BLEU scores of over 0.7 and low error rates throughout. The results for out-of-domain training are lower, as can be expected, given that no data from the target domains was used. The referring expression domain is an exception to this, with the target domain REFCOCO achieving equivalent scores to the source domain GRE. Closer inspection suggests that a reason might be the lower number of spatial relations in REFCOCO in contrast to GRE, so that the majority of noun phrase structures could readily be transferred across the two domains. GRE trained from GIVE similarly does better than GRE alone. A reason for this is presumably the higher amount of spatial relations in GIVE which leads to a better and more balanced representation being learnt than in GRE alone.

A further interesting observation is that for both referring expressions and navigation, learning from an out-of-domain prior achieves better results than learning from a single domain only. This seems to suggest that the pre-learnt weights from a similar dataset are very valuable for the new domains. This is particularly interesting for domains for which little training data is available. We can also see from Table II that SFXH achieves very decent performance *without any in-domain data* at all, but based purely on training from SFXR. This is a remarkable result because it means that we can generate inputs for a new domain based on no annotated training data at all. These results clearly show the significance of using a common input representation across domains. Abstracting away from particular slots, such as "Kirin restaurant" or "Pacific Heights area", we can reuse the lexical-syntactic patterns learnt in one domain in others. Table III shows examples of the abstract patterns and realizations that were transferred across domains.

Comparing with related work, Yu et al. [72] reported a BLEU-1 score of 0.59 and a BLEU-2 score of 0.39 for REFCOCO. This is substantially lower than our scores and might reflect the increased difficulty in the scenario in Yu et al. [72] who generated referring expressions directly from images. In terms of navigation, most related work on the SAIL data [24]–[26] focuses on generating action sequences rather

TABLE II

OBJECTIVE RESULTS IN TERMS OF BLEU-3, BLEU-4 AND SEMANTIC ERROR, AND SUBJECTIVE RESULTS ON THE RATING OF THE NATURALNESS OF UTTERANCES (AVERAGES ARE SHOWN ALONGSIDE MEDIANS IN PARENTHESES). RESULTS ARE SHOWN FOR IN-DOMAIN TRAINING, OUT-OF-DOMAIN TRAINING AND TRAINING FROM PRIOR KNOWLEDGE . SYMBOL * INDICATES STATISTICAL SIGNIFICANCE BETWEEN OUT-OF-DOMAIN TRAINING AND TRAINING WITH PRIOR KNOWLEDGE.

	SYSTEM	BLEU-3	BLEU-4	SEM ERR	NATURALNESS
Human	GRE-HUMAN	1.0	1.0	0.0	2.97 (4)
	REFCOCO	1.0	1.0	0.0	3.43 (4)
	GIVE	1.0	1.0	0.0	3.77 (4)
	SAIL	1.0	1.0	0.0	4.36 (5)
	SFXR	1.0	1.0	0.0	4.17 (4)
	SFXH	1.0	1.0	0.0	3.93 (4)
in-domain	GRE-HUMAN	0.90	0.88	0.016	3.12 (3)
	REFCOCO	0.88	0.82	0.02	3.45 (4)
	GIVE	0.79	0.78	0.09	2.76 (2)
	SAIL	0.82	0.77	0.04	4.02 (4)
	SFXR	0.81	0.76	0.10	3.67 (4)
	SFXH	0.75	0.69	0.08	3.95 (4)
out-of-domain	GRE trained from GIVE	0.94	0.92	0.04	3.11 (3)
	REFCOCO trained from GRE	0.91	0.84	0.02	3.20 (3)
	GIVE trained from GRE	0.23	0.16	0.29	2.03 (2)
	SAIL trained from GIVE	0.68	0.63	0.10	2.95 (3)
	SFXH trained from SFXR	0.61	0.51	0.12	3.45 (3)
prior	GRE with GIVE prior	0.99	0.98	0.0	3.29 (3)
	REFCOCO with GRE prior	0.96	0.77	0.01	3.41 (4)
	GIVE with GRE prior	0.89	0.87	0.06	3.09 (3) *
	SAIL with GIVE prior	0.80	0.72	0.05	3.44 (4) *
	SFXH with SFXR prior	0.74	0.56	0.08	3.58 (4)

than the actual route instructions. For spoken dialogue, Wen et al. [34] achieve BLEU-4 scores 0.73 and 0.83 for SFXR and SFXH, respectively, with a semantic error rate of 0.046. While these results are slightly better than ours for in-domain training, our models are able to transfer weights and train a reasonable policy for SFXH from SFXR alone.

The most relevant comparison to our model in terms of multi-task learning is Wen et al. [12]. The authors achieved a maximum BLEU score of 0.48 for domain transferred data, only 52% of our 0.92, but they worked on the TV and laptop domain, thus not allowing a direct comparison between results. The authors reported a semantic error of 0.04. In contrast to our work, which transfers learnt models across domains and natural data, Wen et al.'s experiments are based on artificially generated data. These often do not display the same variety and complexity as naturally occurring data, which arguably shows that our AMR-based inputs allow for more complex lexical-syntactic constructions to be learnt.

C. Results: Subjective Evaluation

To also assess the subjective quality of generated outputs, we recruited 204 human judges from the CrowdFlower[2] and AMT[3] crowdsourcing platforms to assign subjective ratings to generated outputs. All judges were self-declared native or fluent speakers of English and rated altogether 3425 utterances sampled randomly from a pool of 120 candidates per model. To allow for a comparison with related work, we follow previous authors in asking judges to rate the *naturalness* of utterances. They were asked to agree with the statement *"The utterance is natural (i.e. could have been produced by a human)."* on a scale of 1-5, where 1 is the worst score and 5 is the best. For each dataset, we also collected an equal number of ratings for the original human utterances to provide an upper bound for the comparison of our systems. The results are shown in the right-most column in Table II. Medians are shown alongside averages in parentheses. In a statistical analysis, we decided to focus on the difference between out-of-domain training

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

[3]https://www.mturk.com

TABLE III
EXAMPLES OF LEXICAL-SYNTACTIC PATTERNS LEARNT IN ONE DOMAIN THEN USED IN ANOTHER.

Imperative clause construction (with relative clause)

```
(e1 / event :arg0 (y / you) :arg1 (b1 / obj :mod property :location (w / obj :op1 (o1
                          / on [in]) )) :mode imperative )
```

GIVE: "Click the red button (that is) on the wall."
SFXR: "Try Chinese restaurant Kirin in the Pacific Heights area."

Transitive clause construction

```
(e1 / event :arg0 (b1 / obj :mod property) :arg1 (b2 / obj :mod property))
```

GRE: "The yellow sphere (that is) touching the blue box."
SFXR: "Source restaurant serves Italian food."

Complex noun phrase and spatial relation and temporal adverb

```
(e1 / obj :time (n / now) :domain (b1 / obj :mod property :mod property :location (l /
                          obj :op1 ( n / on [near, by]))) )
```

GRE: "Now, the blue circle on the green square."
GRE: "Now, the green button by the window."
SFXR: "Now, an Indian restaurant near Pacific Heights."

and training with prior knowledge to see what effects can be gained by having prior weights for training. Symbol * indicates statistical significance at $p<0.05$ according to a 2-tailed Wilcoxon signed rank test. From the analysis we can see that two of the six comparisons are statistically significant, namely GIVE with GRE prior and SAIL with GIVE prior—both in the navigation domain. None of the other differences are significant. We believe that these results are encouraging in that we did not expect all differences to be statistically significant. For example, while no significance between in-domain and out-of-domain training or with prior knowledge does of course not indicate equivalent policies or performance, it means at least that the transfer of training data from one domain to another does not lead to a significant deterioration of generated outputs.

The overall results correspond to the objective results. While most of the subjective ratings are not as good as those received by the human utterances (GRE is an exception here), in-domain training from out-of-domain prior knowledge achieves better ratings than pure in-domain training. For domains with sufficient similarity, out-of-domain training achieves equivalent ratings to in-domain training. This further confirms that domain transfer is possible using sufficiently expressive input representations and a generalizable learning model.

To compare with related work on GIVE, Benotti and Denis [73] reported a naturalnesse score of 3.2 for a corpus-based selection method. Denis et al. [74] reported a naturalness score of 2.4 for a system that relies on a linguistically-inspired algorithm for generating referring expressions, and Dethlefs and Cuayáhuitl [9] reported a naturalness score of 3.36 for a system based on hierarchical reinforcement learning. For the SFX datasets, Wen et al. [34] reported a naturalness score of 4.18 for both SFXR and SFXH for in-domain training.

VI. CONCLUSION

We presented an LSTM encoder-decoder model that learns a mapping between a sequence of semantic input symbols and a sequence of words. In comparison to previous work on this topic, we propose the use of AMRs as a common input representation to allow for the transfer of learnt models across domains. Experiments in three different training scenarios show that our model can achieve up to $75-100\%$ of the performance of in-domain training when transferring from a source domain with sufficient training data to

a target domain with no training data at all—provided that the domains are similar in their lexical-syntactic patterns. In-domain training is consistently improved by an out-of-domain prior. We made the following contributions in this research.

1) We present an application of a sequence-to-sequence model to NLG in multiple heterogeneous domains—previous work has so far focused on single domains or domains that were closely related.
2) We use AMRs as inputs to the sequence-to-sequence model as a common input representation that is expressive and transferable across domains. This allows general lexical-syntactic patterns to be learnt that are independent of domain-specific semantic slots.
3) We show that using a generalizable input representation, out-of-domain training can achieve results that are equivalent to in-domain training—or even better in one domain. Training a model based on an appropriate prior model from a source domain can outperform in-domain training by up to 10%. These results show that it is possible to train acceptable models for NLG for new domains with no training data at all.
4) All our code and data are available as resources to the research community.

Future work can investigate metrics to automatically measure the similarity between domains to identify suitable source domains for new target domains. Using semantic embeddings as an input representation would likely further improve the cross-domain results over the 1-hot vector representation we use in this work. We would also like to continue work on NLG for domains in which datasets of unlabelled examples are available by using statistical models learnt for other domains. Finally, several pieces of related work have demonstrated the benefit of using an attention mechanism in NLG with RNNs, particularly for work that deals with longer sequences [35], [37], [58], [75]. While in this work many of our generated output sequences were short, and we therefore decided to focus on the contribution of AMRs, future work should experiment with an attention mechanism to observe potential benefits.

ACKNOWLEDGEMENTS

We acknowledge the VIPER high-performance computing facility of the University of Hull and its support team. We are also grateful for Nvidia's donation of a Titan X Pascal graphics card for our work on deep learning.

REFERENCES

[1] E. Reiter and R. Dale, *Building Natural Language Generation Systems*. New York, NY, USA: Cambridge University Press, 2000.
[2] M. White, R. Rajkumar, and S. Martin, "Towards broad coverage surface realization with CCG," in *Proceedings of the Workshop on Using Corpora for NLG: Language Generation and Machine Translation (UCNLG+MT)*, Copenhagen, Denmark, September 2007, pp. 22–30.
[3] M. Walker, A. Stent, F. Mairesse, and R. Prasad, "Individual and domain adaptation in sentence planning for dialogue," *Journal of Artificial Intelligence Research*, vol. 30, pp. 413–456, September-December 2007.
[4] W. Lu, H. T. Ng, and W. S. Lee, "Natural language generation with tree conditional random fields," in *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, August 2009, pp. 400–409.
[5] F. Mairesse, F. Jurčíček, S. Keizer, B. Thomson, K. Yu, and S. Young, "Phrase-based statistical language generation using graphical models and active learning," in *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics (ACL)*, Uppsala, Sweden, August 2010, pp. 1552–1561.
[6] I. Konstas and M. Lapata, "Unsupervised concept-to-text generation with hypergraphs," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Montreal, Canada, June 2012, pp. 752–761.
[7] N. Dethlefs, H. Hastie, H. Cuayáhuitl, and O. Lemon, "Conditional random fields for responsive surface realisation using global features," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sofia, Bulgaria, August 2013, pp. 1254–1263.
[8] N. Dethlefs, H. Hastie, V. Rieser, and O. Lemon, "Optimising incremental dialogue decisions using information density for interactive systems," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-CoNLL)*, Jeju, South Korea, July 2012, pp. 82–93.
[9] N. Dethlefs and H. Cuayáhuitl, "Hierarchical reinforcement learning for situated natural language generation," *Natural Language Engineering*, vol. 21, pp. 391–435, May 2015.
[10] G. Angeli, P. Liang, and D. Klein, "A simple domain-independent probabilistic approach to generation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Cambridge, Massachusetts, October 2010, pp. 502–512.
[11] D. DeVault, D. Traum, and R. Artstein, "Practical grammar-based NLG from examples," in *Proceedings of the International Natural Language Generation Conference (INLG)*, Salt Fork, Ohio, USA, July 2008, pp. 77–85.

[12] T.-H. Wen, M. Gašić, N. Mrkšić, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, and S. Young, "Multi-domain neural network language generation for spoken dialogue systems," in *Proceedings of the 2016 Conference on North American Chapter of the Association for Computational Linguistics (NAACL)*, San Diego, USA, June 2016, pp. 120–129.

[13] R. Socher, C. C.-Y. Lin, A. Y. Ng, and C. D. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, Bellevue, Washington, USA, June-July 2011, pp. 129–136.

[14] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, USA, October 2013, pp. 1044–1054.

[15] R. Socher, J. Pennington, E. Huang, A. Ng, and C. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Edinburgh, Scotland, July 2011, pp. 151–161.

[16] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, "Recurrent conditional random field for language understanding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 4077–4081.

[17] M. Yazdani and J. Henderson, "A model of zero-shot learning of spoken language understanding," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, September 2015, pp. 244–249.

[18] X. Lei, H. Lin, and G. Heigold, "Deep neural networks with auxiliary gaussian mixture models for real-time speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 7634–7638.

[19] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October 2014, pp. 1724–1734.

[20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, Montréal, Canada, December 2014, pp. 3104–3112.

[21] P. Liang, M. Jordan, and D. Klein, "Learning semantic correspondences with less supervision," in *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL)*, Singapore, August 2009, pp. 91–99.

[22] H. Cuayáhuitl, N. Dethlefs, H. Hastie, and X. Liu, "Training a statistical surface realiser from automatic slot labelling," in *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT)*, South Lake Tahoe, USA, December 2014, pp. 112–117.

[23] B. Snyder and R. Barzilay, "Database-text alignment via structured multilabel classication," in *Proceedings of 20th International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, January 2007, pp. 1713–1718.

[24] D. Chen and R. Mooney, "Learning to interpret natural language navigation instructions from observations," in *Proceedings of 25th AAAI Conference on Artificial Intelligence (AAAI)*, San Francisco, CA, USA, August 2011, pp. 859–865.

[25] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language knowledge, and action in route instructions," in *Proceedings of National Conference on Artificial Intelligence (AAAI)*, Boston, Massachusetts, July 2006, pp. 1475–1482.

[26] H. Mei, M. Bansal, and M. Walter, "Listen, attend and walk: Neural mapping of navigational instructions to action sequences," in *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, Phoenix, Arizona, USA, February 2016, pp. 2772–2778.

[27] A. F. Daniele, M. Bansal, and M. R. Walter, "Navigational instruction generation as inverse reinforcement learning with neural machine translation," in *Proceedings of the Conference on Human-Robot Interaction (HRI)*, Vienna, Austria, March 2017, pp. 109–118.

[28] S. Oswald, H. Kretzschmar, W. Burgard, and C. Stachniss, "Learning to give route directions from human demonstrations," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May-June 2014, pp. 3303–3308.

[29] M. Iyyer, J. Boyd-Graber, and H. Daumé III, "Generating sentences from semantic vector space representations," in *Proceedings of NIPS Workshop on Learning Semantics*, December 2014, pp. 1–5.

[30] J. Andreas and Z. Ghahramani, "A generative model of vector space semantics," in *Proceedings of the ACL Workshop on Continuous Vector Space Models and their Compositionality*, Sofia, Bulgaria, August 2013, pp. 91–99.

[31] G. Dinu and M. Baroni, "How to make words with vectors: Phrase generation in distributional semantics," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Baltimore, Maryland, June 2014, pp. 624–633.

[32] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of INTERSPEECH*, Makuhari, Chiba, Japan, September 2010.

[33] I. Sutskever, J. Martens, and G. Hinton, "Generating text with recurrent neural networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, Bellevue, Washington, USA, June-July 2011, pp. 1017–1024.

[34] T.-H. Wen, M. Gašić, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young, "Semantically conditioned LSTM-based natural language generation for spoken dialogue systems," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, September 2015, pp. 1711–1721.

[35] O. Dusek and F. Jurcicek, "Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany, August 2016, pp. 45–51.

[36] J. Chung, c. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proceedings of the NIPS workshop on Deep Learning*, Montréal, Canada, December 2014, pp. 1–9.

[37] S. Chopra, M. Auli, and A. Rush, "Abstractive sentence summarization with attentive neural networks," in *Proceedings of the Conference on North American Chapter of the Association for Computational Linguistics (NAACL)*, San Diego, USA, June 2016, pp. 93–98.

[38] S. Gerani, G. Carenini, and R. Ng, "Modeling content and structure for abstractive review summarization," *Computer Speech & Language*, in press.

[39] H. Mei, M. Bansal, and M. Walter, "What to talk about and how? Selective generation using LSTMs with coarse-to-fine alignment," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, San Diego, USA, June 2016, pp. 720–730.

[40] C. Chelba and A. Acero, "Adaptation of maximum entropy capitalizer: Little data can help a lot," *Computer Speech & Language*, vol. 20, no. 4, pp. 382–399, October 2006.

[41] H. Daumé-III, "Frustratingly easy domain adaptation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Columbus, Ohio, USA, June 2007, pp. 256–263.

[42] H. Daumé-III and J. Jagarlamudi, "Domain adaptation for machine translation by mining unseen words," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, Portland, Oregon, USA, June 2011, pp. 407–412.

[43] Y. Kim, K. Stratos, R. Sarikaya, and M. Jeong, "New transfer learning techniques for disparate label sets," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Beijing, China, July 2015, pp. 473–482.

[44] A. Jaech, L. Heck, and M. Ostendorf, "Domain adaptation of recurrent neural networks for natural language understanding," in *INTERSPEECH*, San Francisco, USA, September 2016, pp. 690–694.

[45] H. Cuayáhuitl, S. Yu, A. Williamson, and J. Carse, "Deep reinforcement learning for multi-domain dialogue systems," in *Proceedings of the NIPS Workshop on Deep Reinforcement Learning*, Barcelona, Spain, December 2016, pp. 1–9.

[46] ——, "Scaling up deep reinforcement learning for multi-domain dialogue systems," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Anchorage, Alaska, USA, May 2017.

[47] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, March 1994.

[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, November 1997.

[49] A. Graves, "Generating sequences with recurrent neural networks," *Computing Research Repository*, vol. abs/1308.0850, 2013. [Online]. Available: http://arxiv.org/abs/1308.0850

[50] J. Viethen and R. Dale, "Gre3d7: A corpus of distinguishing descriptions for objects in visual scenes," in *Proceedings of the Language Generation and Evaluation Workshop (UCNLG+Eval)*, Edinburgh, Scotland, July 2011, pp. 12–22.

[51] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "Referitgame: Referring to objects in photographs of natural scenes," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October 2014, pp. 787–798.

[52] A. Gargett, K. Garoufi, A. Koller, and K. Striegnitz, "The GIVE-2 corpus of generating instructions in virtual environments," in *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Malta, May 2010, pp. 1–6.

[53] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, "Abstract meaning representation for sembanking," in *Proceedings of the Linguistic Annotation Workshop (LAW)*, Sofia, Bulgaria, August 2013, pp. 178–186.

[54] J. Flanigan, S. Thomson, J. Carbonell, C. Dyer, and N. Smith, "A discriminative graph-based parser for the abstract meaning representation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Baltimore, USA, June 2014, pp. 1426–1436.

[55] S. Misra and Y. Artzi, "Neural shift-reduce CCG semantic parsing," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, November 2016, pp. 1775–1786.

[56] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, "Toward abstractive summarization using semantic representations," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Denver, Colorado, US, May-June 2015, pp. 1077–1086.

[57] J. Flanigan, C. Dyer, N. A. Smith, and J. Carbonell, "Generation from abstract meaning representation using tree transducers," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, San Diego, CA, USA, June 2016, pp. 731–739.

[58] Y. Chen, D. Hakkani-Tur, G. Tur, A. Celikyilmaz, J. Gao, and L. Deng, "Knowledge as a teacher: Knowledge-guided structural attention networks," Tech. Rep. Arxiv report 1609.03286, 2016.

[59] J. Bateman, "Enabling technology for multilingual natural language generation: The KPML development environment," *Natural Language Engineering*, vol. 3, no. 1, pp. 1–42, March 1997.

[60] I. Mel'cuk, "Meaning-text models: A recent trend in Soviet linguistics," *Annual Review of Anthropology*, no. 10, pp. 27–62, 1981.

[61] I. Mel'cuk and A. Polguere, "A formal lexicon in the meaning-text theory (or how to do lexica with words)," *Computational Linguistics*, vol. 13, no. 3-4, July-December 1987.

[62] R. Kittredge, L. Iordanskaj, and A. Polguere, "Multilingual text generation and the meaning-text theory," in *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Pittsburgh, PA, USA, 1988, pp. 1–13.

[63] B. Lavoie and O. Rambow, "A fast and portable realizer for text generation systems," in *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLC)*, Washington, DC, USA, March-April 1997, pp. 265–268.

[64] A. Gatt and E. Reiter, "SimpleNLG: A realisation engine for practical applications," in *Proceedings of the European Workshop on Natural Language Generation (ENLG)*, Athens, Greece, March 2009, pp. 90–93.

[65] M. Bollmann, "Adapting SimpleNLG to German," in *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, Nancy, France, June 2011, pp. 133–138.

[66] P.-L. Vaudry and G. Lapalme, "Adapting SimpleNLG for bilingual English-French realisation," in *Proceedings of the 14th European Workshop on Natural Language Generation (ENLG)*, Sofia, Bulgaria, August 2013, pp. 183–187.

[67] A. Mazzei, C. Battaglino, and C. Bosco, "SimpleNLG-IT: Adapting SimpleNLG to Italian," in *Proceedings of the 9th International Natural Language Generation Conference (INLG)*, Edinburgh, Scotland, September 2016, pp. 184–192.

[68] K. Toutanova, D. Klein, C. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the North American Chapter of the Annual Meeting of the Association for Computational Linguistics (NAACL)*, Edmonton, Canada, May-June 2003, pp. 173–180.

[69] F. Chollet, "Keras," https://github.com/fchollet/keras, 2016.

[70] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[71] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, Toulouse, France, July 2001, pp. 311–318.
[72] L. Yu, P. Poirson, S. Yang, A. Berg, and T. Berg, "Modeling context in referring expressions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, October 2016, pp. 69–85.
[73] L. Benotti and A. Denis, "Giving instructions in virtual environments by corpus-based selection," in *Proceedings of the 12th Annual Meeting on Discourse and Dialogue (SIGdial)*, Portland, OR, USA, June 2011, pp. 68–77.
[74] A. Denis, M. Amoia, L. Benotti, L. Perez-Beltrachini, C. Gardent, and T. Osswald, "The GIVE-2 Nancy generation systems NA and NM," in *Proceedings of the International Natural Language Generation Conference (INLG)*, Dublin, Ireland, July 2010, pp. 1–12.
[75] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.