Exploiting Semantics in Neural Machine Translation with Graph Convolutional Networks

Diego Marcheggiani \textsuperscript{1,2}, Jasmijn Bastings \textsuperscript{2} and Ivan Titov \textsuperscript{1,2}

\textsuperscript{1}ILCC, School of Informatics, University of Edinburgh
\textsuperscript{2}ILLC, University of Amsterdam

marcheggiani@uva.nl bastings@uva.nl titov@inf.ed.ac.uk

Abstract

Semantic representations have long been argued as potentially useful for enforcing meaning preservation and improving generalization performance of machine translation methods. In this work, we are the first to incorporate information about predicate-argument structure of source sentences (namely, semantic-role representations) into neural machine translation. We use Graph Convolutional Networks (GCNs) to inject a semantic bias into sentence encoders and achieve improvements in BLEU scores over the linguistic-agnostic and syntax-aware versions on the English–German language pair.

1 Introduction

It has long been argued that semantic representations may provide a useful linguistic bias to machine translation systems (Weaver, 1955; Bar-Hillel, 1960). Semantic representations provide an abstraction which can generalize over different surface realizations of the same underlying ‘meaning’. Providing this information to a machine translation system, can, in principle, improve meaning preservation and boost generalization performance.

Though incorporation of semantic information into traditional statistical machine translation has been an active research topic (e.g., (Baker et al., 2012; Liu and Gildea, 2010; Wu and Fung, 2009; Bazrafshan and Gildea, 2013; Aziz et al., 2011; Jones et al., 2012)), we are not aware of any previous work considering semantic structures in neural machine translation (NMT). In this work, we aim to fill this gap by showing how information about predicate-argument structure of source sentences can be integrated into standard attention-based NMT models (Bahdanau et al., 2015).

We consider PropBank-style (Palmer et al., 2005) semantic role structures, or more specifically their dependency versions (Surdeanu et al., 2008). The semantic-role representations mark semantic arguments of predicates in a sentence and categorize them according to their semantic roles. Consider Figure 1, the predicate gave has three arguments: \textsuperscript{1} John (semantic role \textsuperscript{A0}, ‘the giver’), \textsuperscript{2} wife (\textsuperscript{A2}, ‘an entity given to’) and \textsuperscript{3} present (\textsuperscript{A1}, ‘the thing given’). Semantic roles capture commonalities between different realizations of the same underlying predicate-argument structures. For example, present will still be \textsuperscript{A1} in sentence “John gave a nice present to his wonderful wife”, despite different surface forms of the two sentences. We hypothesize that semantic roles can be especially beneficial in NMT, as ‘argument switching’ (flipping arguments corresponding to different roles) is one of frequent and severe mistakes made by NMT systems (Isabelle et al., 2017).

There is a limited amount of work on incorporating graph structures into neural sequence models. Though, unlike semantics in NMT, syntactically-aware NMT has been a relatively hot topic recently, with a number of approaches claiming improvements from using treebank syntax (Sennrich and Haddow, 2016; Eriguchi et al., 2016; Nadejde et al., 2017; Bastings et al., 2017; Aharoni and Goldberg, 2017), our graphs are different from syntactic structures. Unlike syntactic dependency graphs, they are not trees and thus

\textsuperscript{1}We slightly abuse the terminology: formally these are syntactic heads of arguments rather than arguments.
cannot be processed in a bottom-up fashion as in 
Eriguchi et al. (2016) or easily linearized as in 
Aharoni and Goldberg (2017). Luckily, the model-
ing approach of Bastings et al. (2017) does not make any assumptions about the graph structure, and thus we build on their method.

Bastings et al. (2017) used Graph Convolutional Networks (GCNs) to encode syntactic structure. GCNs were originally proposed by Kipf and Welling (2016) and modified to handle labeled and automatically predicted (hence noisy) syntactic dependency graphs by Marcheggiani and Titov (2017). Representations of nodes (i.e. words in a sentence) in GCNs are directly influenced by representations of their neighbors in the graph. The form of influence (e.g., transition matrices and parameters of gates) are learned in such a way as to benefit the end task (i.e. translation). These linguistically-aware word representations are used within a neural encoder. Although recent research has shown that neural architectures are able to learn some linguistic phenomena without explicit linguistic supervision (Linzen et al., 2016; Vaswani et al., 2017), informing word representations with linguistic structures can provide a useful inductive bias.

We apply GCNs to the semantic dependency graphs and experiment on the English–German language pair (WMT16). We observe an improvement over the semantics-agnostic baseline (a BiRNN encoder; 23.3 vs 24.5 BLEU). As we use exactly the same modeling approach as in the syntactic method of Bastings et al. (2017), we can easily compare the influence of the types of linguistic structures (i.e., syntax vs. semantics). We observe that when using full WMT data we obtain better results with semantics than with syntax (23.9 BLEU for syntactic GCN). Using syntactic and semantic GCNs together, we obtain a further gain (24.9 BLEU) which suggests the complementarity of information encoded by the syntactic and semantic representations.

2 Model

2.1 Encoder-decoder Models

We use a standard attention-based encoder-decoder model (Bahdanau et al., 2015) as a starting point for constructing our model. In encoder-decoder models, the encoder takes as input the source sentence \( x \) and calculates a representation of each word \( x_t \) in \( x \). The decoder outputs a translation \( y \) relying on the representations of the source sentence. Traditionally, the encoder is parametrized as a Recurrent Neural Network (RNN), but other architectures have also been successful, such as Convolutional Neural Networks (CNN) (Gehring et al., 2017) and hierarchical self-attention models (Vaswani et al., 2017), among others. In this paper we experiment with RNN and CNN encoders. We explore the benefits of incorporating information about semantic-role structures into such encoders.

More formally, RNNs (Elman, 1990) can be defined as a function \( RNN(x_{1:t}) \) that calculates the hidden representation \( h_t \) of a word \( x_t \) based on its left context. Bidirectional RNNs use two RNNs: one runs in the forward direction and another one in the backward direction. The forward \( RNN(x_{1:t}) \) represents the left context of word \( x_t \), whereas the backward \( RNN(x_{n:t}) \) computes a representation of the right context. The two representations are concatenated in order to incorporate information about the entire sentence:

\[
\text{BiRNN}(x, t) = RNN(x_{1:t}) \circ RNN(x_{n:t}).
\]

In contrast to BiRNNs, CNNs (LeCun et al., 2001) calculate a representation of a word \( x_t \) by considering a window of words \( w \) around \( x_t \), such as

\[
\text{CNN}(x, t, w) = f(x_{t-w/2}, x_{t}, \ldots, x_{t+w/2}),
\]

where \( f \) is usually an affine transformation followed by a nonlinear function.

Once the sentence has been encoded, the decoder takes as input the induced sentence representation and generates the target sentence \( y \). The target sentence \( y \) is predicted word by word using an RNN decoder. At each step, the decoder calculates the probability of generating a word \( y_t \) conditioning on a context vector \( c_t \) and the previous state of the RNN decoder. The context vector \( c_t \) is calculated based on the representation of the source sentence computed by the encoder, using an attention mechanism (Bahdanau et al., 2015).

Such a model is trained end-to-end on a parallel corpus to maximize the conditional likelihood of the target sentences.

2.2 Graph Convolutional Networks

Graph neural networks are a family of neural architectures (Scarselli et al., 2009; Gilmer et al., 2017) specifically devised to induce representation of nodes in a graph relying on its graph struc-
are particularly useful when the graph is predicted and thus may contain errors, i.e., wrong edges. In this scenario gates can down weight the influence of potentially unreliable edges. The function $\rho$ is a non-linearity (ReLU).\footnote{Refer to Marcheggiani and Titov (2017) and Bastings et al. (2017) for further details.}

As with CNNs, GCN layers can be stacked in order to incorporate higher order neighborhoods. In our experiments, we used GCNs on top of a standard BiRNN encoder and a CNN encoder (Figure 2). In other words, the initial representations of words fed into GCN were either RNN states or CNN representations.

### 3 Experiments

We experimented with the English-to-German WMT16 dataset (~4.5 million sentence pairs for training). We use its subset, News Commentary v11, for development and additional experiments (~226,000 sentence pairs). For all these experiments, we use newstest2015 and newstest2016 as a validation and test set, respectively.

We parsed the English partitions of these datasets with a syntactic dependency parser (Andor et al., 2016) and dependency-based semantic role labeler (Marcheggiani et al., 2017). We constructed the English vocabulary by taking all words with frequency higher than three, while for German we used byte-pair encodings (BPE) (Sennrich et al., 2016). All hyperparameter selection was performed on the validation set (see Appendix A). We measured the performance of the models with (cased) BLEU scores (Papineni et al., 2002). The settings and the framework (Neural Monkey (Helcl and Libovický, 2017)) used for experiments are the ones used in Bastings et al. (2017), which we use as baselines. As RNNs, we use GRUs (Cho et al., 2014).

We now discuss the impact that different architectures and linguistic information have on the translation quality.

<table>
<thead>
<tr>
<th></th>
<th>BiRNN</th>
<th>CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Bastings et al., 2017)</td>
<td>14.9</td>
<td>12.6</td>
</tr>
<tr>
<td>+Sem</td>
<td>15.6</td>
<td>13.4</td>
</tr>
<tr>
<td>+Syn (Bastings et al., 2017)</td>
<td>16.1</td>
<td>13.7</td>
</tr>
<tr>
<td>+Syn + Sem</td>
<td>15.8</td>
<td>14.3</td>
</tr>
</tbody>
</table>

Table 2: Test BLEU, En–De, News Commentary.
3.1 Results and Discussion

First, we start with experiments with the smaller News Commentary training set (see Table 1). As in Bastings et al. (2017), we used the standard attention-based encoder-decoder model as a baseline.

We tested the impact of semantic GCNs when used on top of CNN and BiRNN encoders. As expected, BiRNN results are stronger than CNN ones. In general, for both encoders we observe the same trend: using semantic GCNs leads to an improvement over the baseline model. The improvements is 0.7 BLEU for BiRNN and 0.8 for CNN. This is slightly surprising as the potentially non-local semantic information should in principle be more beneficial within a less powerful and local CNN encoder. The syntactic GCNs (Bastings et al., 2017) appear stronger than semantic GCNs. As exactly the same model and optimization are used for both GCNs, the differences should be due to the type of linguistic representations used. Since syntax and semantic structures seem to be individually beneficial and, though related, capture different linguistic phenomena, it is natural to try combining them. When syntactic and semantic GCNs are stacked, we observe a further improvement with respect to the semantic GCN model, and a substantial improvement with respect to the syntactic GCN model with a CNN encoder.

Now we turn to the full WMT experiments. Though we expected the linguistic bias to be more valuable in the resource-poor setting, the improvement from using semantic-role structures is larger here (+1.2 BLEU). It is surprising but perhaps more data is beneficial for accurately modeling influence of semantics on the translation task. Interestingly, the syntactic GCN now outperforms the syntactic one by 0.6 BLEU. Again, it is hard to pinpoint exact reasons for this. One may speculate though that, given enough data, RNNs were able to capture syntactic dependency and thus reducing the benefits from using treebank syntax, whereas (often less local and harder) semantic dependencies were more complementary. Finally, when the semantic GCN is stacked over the syntactic GCN, we obtain a further improvement reaching 24.9 BLEU. These results suggest that syntactic and semantic dependency structures are complementary information when it comes to translation.

3.2 Ablations and Syntax-Semantics GCNs

We used the validation set to perform extra experiments, as well as to select hyper parameters (e.g., the number of GCN layers) for the experiments presented above. Table 3 presents the results. The annotation 1L, 2L and 3L refers to the number of GCN layers used.

First, we tested whether the gain we observed is an effect of an extra layer of non-linearity or an effect of the linguistic structures encoded with GCNs. In order to do so, we used the GCN layer without any structural information. In this way, only the self-loop edge is used within the GCN node updates. These results (e.g., BiRNN+SelfLoop) show that the linguistic-agnostic GCNs perform on par with the baseline, and thus using linguistic structure is genuinely beneficial in translation.

We compared stacking a semantic GCN on top of syntactic one (as done in section 3.1) against combining syntax and semantic in the same GCN layer (SemSyn). With SemSyn we do not observe any improvement with respect to having semantic and syntactic information alone. We argue that the reason for this is that the two linguistic representations do not interact much as much as needed when encoded into the same GCN layer with a simple aggregation function. The stacking approach allows for more complex interaction and more successful. However, on this smaller dataset,

---

Note that the SRL system we use (Marcheggiani et al., 2017) does not use syntax and is faster than the syntactic parser of Andor et al. (2016), so semantic GCNs may still be preferable from the engineering perspective even in this setting.

---

We used distinct matrices W for syntax and semantics.

---

<table>
<thead>
<tr>
<th></th>
<th>BiRNN</th>
<th>CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Bastings et al., 2017)</td>
<td>14.1</td>
<td>12.1</td>
</tr>
<tr>
<td>+Sem (1L)</td>
<td>14.3</td>
<td>12.5</td>
</tr>
<tr>
<td>+Sem (2L)</td>
<td>14.4</td>
<td>12.6</td>
</tr>
<tr>
<td>+Sem (3L)</td>
<td>14.4</td>
<td>12.7</td>
</tr>
<tr>
<td>+Syn (2L) (Bastings et al., 2017)</td>
<td>14.8</td>
<td>13.1</td>
</tr>
<tr>
<td>+SelfLoop (1L)</td>
<td>14.1</td>
<td>12.1</td>
</tr>
<tr>
<td>+SelfLoop (2L)</td>
<td>14.2</td>
<td>11.5</td>
</tr>
<tr>
<td>+SemSyn (1L)</td>
<td>14.1</td>
<td>12.7</td>
</tr>
<tr>
<td>+Syn (1L) + Sem (1L)</td>
<td>14.7</td>
<td>12.7</td>
</tr>
<tr>
<td>+Syn (1L) + Sem (2L)</td>
<td>14.6</td>
<td>12.8</td>
</tr>
<tr>
<td>+Syn (2L) + Sem (1L)</td>
<td>14.9</td>
<td>13.0</td>
</tr>
<tr>
<td>+Syn (2L) + Sem (2L)</td>
<td>14.9</td>
<td>13.5</td>
</tr>
</tbody>
</table>

Table 3: Validation BLEU, News commentary only.
unlike full WMT (Table 1), we obtain smaller improvements over the single-representation models.

3.3 Qualitative Analysis

We analyzed the behavior of the BiRNN baseline and the semantic GCN model trained on the full WMT16 training set. In Table 4 we show three examples where there is a clear difference between translations produced by the two models. Besides the two translations, we show the dependency SRL structure predicted by the labeler and exploited by our GCN model.

In the first sentence, the only difference is in the choice of the preposition for the argument Mark. Note that the argument is correctly assigned to role A2 (‘Buyer’) by the semantic role labeler. The BiRNN model translates \textit{to} with nach, which in German expresses directionality and would be a correct translation should the argument refer to a location. In contrast, semantic GCN correctly translates \textit{to} as an. We hypothesize that the semantic structure, namely the assignment of the argument to A2 rather than AM-DIR (‘Directionality’), helps the model to choose the right preposition.

In the second sentence, the BiRNN’s translation is ungrammatical, whereas semantic GCN is able to correctly translate the source sentence. Again, the arguments, correctly identified by semantic role labeler, may have been useful in translating this somewhat tricky sentence. Finally, in the third case, we can observe that both translations are problematic. BiRNN and Semantic GCN ignored verbs \textit{sit} and \textit{play}, respectively. However, BiRNN’s translation for this sentence is preferable, as it is grammatically correct, even if not fluent or particularly precise.

4 Conclusions

In this work we propose injecting information about predicate-argument structures of sentences in NMT models. We observe that the semantic structures are beneficial for the English–German language pair. So far we evaluated the model performance in terms of BLEU only. It would be interesting in future work to both understand when semantics appears beneficial, and also to see which components of semantic structures play a role. Experiments on other language pairs are also left for future work.

Acknowledgments

We thank Stella Frank and Wilker Aziz for their suggestions and comments. The project was supported by the European Research Council (ERC StG BroadSem 678254), and the Dutch National Science Foundation (NWO VIDI 639.022.518). We thank NVIDIA for donating GPUs used for this research.
References


