Linguistic Input Features Improve Neural Machine Translation

Abstract

Neural machine translation has recently achieved impressive results, while using little in the way of external linguistic information. In this paper we show that the strong learning capability of neural MT models does not make linguistic features redundant; they can be easily incorporated to provide further improvements in performance. We generalize the embedding layer of the encoder in the attentional encoder--decoder architecture to support the inclusion of arbitrary features, in addition to the baseline word feature. We add morphological features, part-of-speech tags, and syntactic dependency labels as input features to English<->German, and English->Romanian neural machine translation systems. In experiments on WMT16 training and test sets, we find that linguistic input features improve model quality according to three metrics: perplexity, BLEU and CHRF3. An open-source implementation of our neural MT system is available, as are sample files and configurations.

Introduction

Neural machine translation has recently achieved impressive results BIBREF0 , BIBREF1 , while learning from raw, sentence-aligned parallel text and using little in the way of external linguistic information. However, we hypothesize that various levels of linguistic annotation can be valuable for neural machine translation. Lemmatisation can reduce data sparseness, and allow inflectional variants of the same word to explicitly share a representation in the model. Other types of annotation, such as parts-of-speech (POS) or syntactic dependency labels, can help in disambiguation. In this paper we investigate whether linguistic information is beneficial to neural translation models, or whether their strong learning capability makes explicit linguistic features redundant.

Let us motivate the use of linguistic features using examples of actual translation errors by neural MT systems. In translation out of English, one problem is that the same surface word form may be shared between several word types, due to homonymy or word formation processes such as conversion. For instance, close can be a verb, adjective, or noun, and these different meanings often have distinct translations into other languages. Consider the following English INLINEFORM0 German example:

For the English source sentence in Example SECREF4 (our translation in Example SECREF5), a neural MT system (our baseline system from Section SECREF4) mistranslates close as a verb, and produces the German verb schließen (Example SECREF6), even though close is an adjective in this sentence, which has the German translation nah. Intuitively, part-of-speech annotation of the English input could disambiguate between verb, noun, and adjective meanings of close.

As a second example, consider the following German INLINEFORM0 English example:

German main clauses have a verb-second (V2) word order, whereas English word order is generally SVO. The German sentence (Example UID7 ; English reference in Example UID8) topicalizes the predicate gefährlich 'dangerous', putting the subject die Route 'the route' after the verb. Our baseline system (Example UID9) retains the original word order, which is highly unusual in English, especially for prose in the news domain. A syntactic annotation of the source sentence could support the attentional encoder-decoder in learning which words in the German source to attend (and translate) first.

We will investigate the usefulness of linguistic features for the language pair German INLINEFORM0 English, considering the following linguistic features:

The inclusion of lemmas is motivated by the hope for a better generalization over inflectional variants of the same word form. The other linguistic features are motivated by disambiguation, as discussed in our introductory examples.

Neural Machine Translation

We follow the neural machine translation architecture by DBLP:journals/corr/BahdanauCB14, which we will briefly summarize here.

The neural machine translation system is implemented as an attentional encoder-decoder network with recurrent neural networks.

The encoder is a bidirectional neural network with gated recurrent units BIBREF3 that reads an input sequence INLINEFORM0 and calculates a forward sequence of hidden states INLINEFORM1 , and a backward sequence INLINEFORM2 . The hidden states INLINEFORM3 and INLINEFORM4 are concatenated to obtain the annotation vector INLINEFORM5 .

The decoder is a recurrent neural network that predicts a target sequence INLINEFORM0 . Each word INLINEFORM1 is predicted based on a recurrent hidden state INLINEFORM2 , the previously predicted word INLINEFORM3 , and a context vector INLINEFORM4 . INLINEFORM5 is computed as a weighted sum of the annotations INLINEFORM6 . The weight of each annotation INLINEFORM7 is computed through an alignment model INLINEFORM8 , which models the probability that INLINEFORM9 is aligned to INLINEFORM10 . The alignment model is a single-layer feedforward neural network that is learned jointly with the rest of the network through backpropagation.

A detailed description can be found in BIBREF0 , although our implementation is based on a slightly modified form of this architecture, released for the dl4mt tutorial. Training is performed on a parallel corpus with stochastic gradient descent. For translation, a beam search with small beam size is employed.

Adding Input Features

Our main innovation over the standard encoder-decoder architecture is that we represent the encoder input as a combination of features BIBREF4 .

We here show the equation for the forward states of the encoder (for the simple RNN case; consider BIBREF0 for GRU): DISPLAYFORM0

where INLINEFORM0 is a word embedding matrix, INLINEFORM1 , INLINEFORM2 are weight matrices, with INLINEFORM3 and INLINEFORM4 being the word embedding size and number of hidden units, respectively, and INLINEFORM5 being the vocabulary size of the source language.

We generalize this to an arbitrary number of features INLINEFORM0 : DISPLAYFORM0

where INLINEFORM0 is the vector concatenation, INLINEFORM1 are the feature embedding matrices, with INLINEFORM2 , and INLINEFORM3 is the vocabulary size of the INLINEFORM4 th feature. In other words, we look up separate embedding vectors for each feature, which are then concatenated. The length of the concatenated vector matches the total embedding size, and all other parts of the model remain unchanged.

Linguistic Input Features

Our generalized model of the previous section supports an arbitrary number of input features. In this paper, we will focus on a number of well-known linguistic features. Our main empirical question is if providing linguistic features to the encoder improves the translation quality of neural machine translation systems, or if the information emerges from training encoder-decoder models on raw text, making its inclusion via explicit features redundant. All linguistic features are predicted automatically; we use Stanford CoreNLP BIBREF5 , BIBREF6 , BIBREF7 to annotate the English input for English INLINEFORM0 German, and ParZu BIBREF8 to annotate the German input for German INLINEFORM1 English. We here discuss the individual features in more detail.

Lemma

Using lemmas as input features guarantees sharing of information between word forms that share the same base form. In principle, neural models can learn that inflectional variants are semantically related, and represent them as similar points in the continuous vector space BIBREF9 . However, while this has been demonstrated for high-frequency words, we expect that a lemmatized representation increases data efficiency; low-frequency variants may even be unknown to word-level models. With character- or subword-level models, it is unclear to what extent they can learn the similarity between low-frequency word forms that share a lemma, especially if the word forms are superficially dissimilar. Consider the following two German word forms, which share the lemma liegen `lie':

liegt `lies' (3.p.sg. present)

läge `lay' (3.p.sg. subjunctive II)

The lemmatisers we use are based on finite-state methods, which ensures a large coverage, even for infrequent word forms. We use the Zmorge analyzer for German BIBREF10 , BIBREF11 , and the lemmatiser in the Stanford CoreNLP toolkit for English BIBREF6 .

Subword Tags

In our experiments, we operate on the level of subwords to achieve open-vocabulary translation with a fixed symbol vocabulary, using a segmentation based on byte-pair encoding (BPE) BIBREF12 . We note that in BPE segmentation, some symbols are potentially ambiguous, and can either be a separate word, or a subword segment of a larger word. Also, text is represented as a sequence of subword units with no explicit word boundaries, but word boundaries are potentially helpful to learn which symbols to attend to, and when to forget information in the recurrent layers. We propose an annotation of subword structure similar to popular IOB format for chunking and named entity recognition, marking if a symbol in the text forms the beginning (B), inside (I), or end (E) of a word. A separate tag (O) is used if a symbol corresponds to the full word.

Morphological Features

For German INLINEFORM0 English, the parser annotates the German input with morphological features. Different word types have different sets of features – for instance, nouns have case, number and gender, while verbs have person, number, tense and aspect – and features may be underspecified. We treat the concatenation of all morphological features of a word, using a special symbol for underspecified features, as a string, and treat each such string as a separate feature value.

POS Tags and Dependency Labels

In our introductory examples, we motivated POS tags and dependency labels as possible disambiguators. Each word is associated with one POS tag, and one dependency label. The latter is the label of the edge connecting a word to its syntactic head, or 'ROOT' if the word has no syntactic head.

On Using Word-level Features in a Subword Model

We segment rare words into subword units using BPE. The subword tags encode the segmentation of words into subword units, and need no further modification. All other features are originally word-level features. To annotate the segmented source text with features, we copy the word's feature value to all its subword units. An example is shown in Figure FIGREF26 .

Evaluation

We evaluate our systems on the WMT16 shared translation task English INLINEFORM0 German. The parallel training data consists of about 4.2 million sentence pairs.

To enable open-vocabulary translation, we encode words via joint BPE BIBREF12 , learning 89500 merge operations on the concatenation of the source and target side of the parallel training data. We use minibatches of size 80, a maximum sentence length of 50, word embeddings of size 500, and hidden layers of size 1024. We clip the gradient norm to 1.0 BIBREF13 . We train the models with Adadelta BIBREF14 , reshuffling the training corpus between epochs. We validate the model every 10000 minibatches via Bleu and perplexity on a validation set (newstest2013).

For neural MT, perplexity is a useful measure of how well the model can predict a reference translation given the source sentence. Perplexity is thus a good indicator of whether input features provide any benefit to the models, and we report the best validation set perplexity of each experiment. To evaluate whether the features also increase translation performance, we report case-sensitive Bleu scores with mteval-13b.perl on two test sets, newstest2015 and newstest2016. We also report chrF3 BIBREF15 , a character n-gram F INLINEFORM0 score which was found to correlate well with human judgments, especially for translations out of English BIBREF16 . The two metrics may occasionally disagree, partly because they are highly sensitive to the length of the output. Bleu is precision-based, whereas chrF3 considers both precision and recall, with a bias for recall. For Bleu, we also report whether differences between systems are statistically significant according to a bootstrap resampling significance test BIBREF17 .

We train models for about a week, and report results for an ensemble of the 4 last saved models (with models saved every 12 hours). The ensemble serves to smooth the variance between single models.

Decoding is performed with beam search with a beam size of 12.

To ensure that performance improvements are not simply due to an increase in the number of model parameters, we keep the total size of the embedding layer fixed to 500. Table TABREF29 lists the embedding size we use for linguistic features – the embedding layer size of the word-level feature varies, and is set to bring the total embedding layer size to 500. If we include the lemma feature, we roughly split the embedding vector one-to-two between the lemma feature and the word feature. The table also shows the network vocabulary size; for all features except lemmas, we can represent all feature values in the network vocabulary – in the case of words, this is due to BPE segmentation. For lemmas, we choose the same vocabulary size as for words, replacing rare lemmas with a special UNK symbol.

2015arXiv151106709S report large gains from using monolingual in-domain training data, automatically back-translated into the source language to produce a synthetic parallel training corpus. We use the synthetic corpora produced in these experiments (3.6–4.2 million sentence pairs), and we trained systems which include this data to compare against the state of the art. We note that our experiments with this data entail a syntactic annotation of automatically translated data, which may be a source of noise. For the systems with synthetic data, we double the training time to two weeks.

We also evaluate linguistic features for the lower-resourced translation direction English INLINEFORM0 Romanian, with 0.6 million sentence pairs of parallel training data, and 2.2 million sentence pairs of synthetic parallel data. We use the same linguistic features as for English INLINEFORM1 German. We follow sennrich-wmt16 in the configuration, and use dropout for the English INLINEFORM2 Romanian systems. We drop out full words (both on the source and target side) with a probability of 0.1. For all other layers, the dropout probability is set to 0.2.

Results

Table TABREF32 shows our main results for German INLINEFORM0 English, and English INLINEFORM1 German. The baseline system is a neural MT system with only one input feature, the (sub)words themselves. For both translation directions, linguistic features improve the best perplexity on the development data (47.3 INLINEFORM2 46.2, and 54.9 INLINEFORM3 52.9, respectively). For German INLINEFORM4 English, the linguistic features lead to an increase of 1.5 Bleu (31.4 INLINEFORM5 32.9) and 0.5 chrF3 (58.0 INLINEFORM6 58.5), on the newstest2016 test set. For English INLINEFORM7 German, we observe improvements of 0.6 Bleu (27.8 INLINEFORM8 28.4) and 1.2 chrF3 (56.0 INLINEFORM9 57.2).

To evaluate the effectiveness of different linguistic features in isolation, we performed contrastive experiments in which only a single feature was added to the baseline. Results are shown in Table TABREF33 . Unsurprisingly, the combination of all features (Table TABREF32) gives the highest improvement, averaged over metrics and test sets, but most features are beneficial on their own. Subword tags give small improvements for English INLINEFORM0 German, but not for German INLINEFORM1 English. All other features outperform the baseline in terms of perplexity, and yield significant improvements in Bleu on at least one test set. The gain from different features is not fully cumulative; we note that the information encoded in different features overlaps. For instance, both the dependency labels and the morphological features encode the distinction between German subjects and accusative objects, the former through different labels (subj and obja), the latter through grammatical case (nominative and accusative).

We also evaluated adding linguistic features to a stronger baseline, which includes synthetic parallel training data. In addition, we compare our neural systems against phrase-based (PBSMT) and syntax-based (SBSMT) systems by BIBREF18 , all of which make use of linguistic annotation on the source and/or target side. Results are shown in Table TABREF34 . For German INLINEFORM0 English, we observe similar improvements in the best development perplexity (45.2 INLINEFORM1 44.1), test set Bleu (37.5 INLINEFORM2 38.5) and chrF3 (62.2 INLINEFORM3 62.8). Our test set Bleu is on par to the best submitted system to this year's WMT 16 shared translation task, which is similar to our baseline MT system, but which also uses a right-to-left decoder for reranking BIBREF19 . We expect that linguistic input features and bidirectional decoding are orthogonal, and that we could obtain further improvements by combining the two.

For English INLINEFORM0 German, improvements in development set perplexity carry over (49.7 INLINEFORM1 48.4), but we see only small, non-significant differences in Bleu and chrF3. While we cannot clearly account for the discrepancy between perplexity and translation metrics, factors that potentially lower the usefulness of linguistic features in this setting are the stronger baseline, trained on more data, and the low robustness of linguistic tools in the annotation of the noisy, synthetic data sets. Both our baseline neural MT systems and the systems with linguistic features substantially outperform phrase-based and syntax-based systems for both translation directions.

In the previous tables, we have reported the best perplexity. To address the question about the randomness in perplexity, and whether the best perplexity just happened to be lower for the systems with linguistic features, we show perplexity on our development set as a function of training time for different systems (Figure FIGREF35). We can see that perplexity is consistently lower for the systems trained with linguistic features.

Table TABREF36 shows results for a lower-resourced language pair, English INLINEFORM0 Romanian. With linguistic features, we observe improvements of 1.0 Bleu over the baseline, both for the systems trained on parallel data only (23.8 INLINEFORM1 24.8), and the systems which use synthetic training data (28.2 INLINEFORM2 29.2). According to Bleu, the best submission to WMT16 was a system combination by qt21syscomb2016. Our best system is competitive with this submission.

Table TABREF37 shows translation examples of our baseline, and the system augmented with linguistic features. We see that the augmented neural MT systems, in contrast to the respective baselines, successfully resolve the reordering for the German INLINEFORM0 English example, and the disambiguation of close for the English INLINEFORM1 German example.

Related Work

Linguistic features have been used in neural language modelling BIBREF4 , and are also used in other tasks for which neural models have recently been employed, such as syntactic parsing BIBREF7 . This paper addresses the question whether linguistic features on the source side are beneficial for neural machine translation. On the target side, linguistic features are harder to obtain for a generation task such as machine translation, since this would require incremental parsing of the hypotheses at test time, and this is possible future work.

Among others, our model incorporates information from a dependency annotation, but is still a sequence-to-sequence model. 2016arXiv160306075E propose a tree-to-sequence model whose encoder computes vector representations for each phrase in the source tree. Their focus is on exploiting the (unlabelled) structure of a syntactic annotation, whereas we are focused on the disambiguation power of the functional dependency labels.

Factored translation models are often used in phrase-based SMT BIBREF21 as a means to incorporate extra linguistic information. However, neural MT can provide a much more flexible mechanism for adding such information. Because phrase-based models cannot easily generalize to new feature combinations, the individual models either treat each feature combination as an atomic unit, resulting in data sparsity, or assume independence between features, for instance by having separate language models for words and POS tags. In contrast, we exploit the strong generalization ability of neural networks, and expect that even new feature combinations, e.g. a word that appears in a novel syntactic function, are handled gracefully.

One could consider the lemmatized representation of the input as a second source text, and perform multi-source translation BIBREF22 . The main technical difference is that in our approach, the encoder and attention layers are shared between features, which we deem appropriate for the types of features that we tested.

Conclusion

In this paper we investigate whether linguistic input features are beneficial to neural machine translation, and our empirical evidence suggests that this is the case.

We describe a generalization of the encoder in the popular attentional encoder-decoder architecture for neural machine translation that allows for the inclusion of an arbitrary number of input features. We empirically test the inclusion of various linguistic features, including lemmas, part-of-speech tags, syntactic dependency labels, and morphological features, into English INLINEFORM0 German, and

English INLINEFORM1 Romanian neural MT systems. Our experiments show that the linguistic features yield improvements over our baseline, resulting in improvements on newstest2016 of 1.5 Bleu for German INLINEFORM2 English, 0.6 Bleu for English INLINEFORM3 German, and 1.0 Bleu for English INLINEFORM4 Romanian.

In the future, we expect several developments that will shed more light on the usefulness of linguistic (or other) input features, and whether they will establish themselves as a core component of neural machine translation. On the one hand, the machine learning capability of neural architectures is likely to increase, decreasing the benefit provided by the features we tested. On the other hand, there is potential to explore the inclusion of novel features for neural MT, which might prove to be even more helpful than the ones we investigated, and the features we investigated may prove especially helpful for some translation settings, such as very low-resourced settings and/or translation settings with a highly inflected source language.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements 645452 (QT21), and 644402 (HimL).