Referenceless Quality Estimation for Natural Language Generation

Ondřej Dušek 1  Jekaterina Novikova 1  Verena Rieser 1

Abstract
Traditional automatic evaluation measures for natural language generation (NLG) use costly human-authored references to estimate the quality of a system output. In this paper, we propose a referenceless quality estimation (QE) approach based on recurrent neural networks, which predicts a quality score for a NLG system output by comparing it to the source meaning representation only. Our method outperforms traditional metrics and a constant baseline in most respects; we also show that synthetic data helps to increase correlation results by 21% compared to the base system. Our results are comparable to results obtained in similar QE tasks despite the more challenging setting.

1. Introduction
Automatic evaluation of natural language generation (NLG) is a complex task due to multiple acceptable outcomes. Apart from manual human evaluation, most recent works in NLG are evaluated using word-overlap-based metrics such as BLEU (Gkatzia & Mahamood, 2015), which compute similarity against gold-standard human references. However, high quality human references are costly to obtain, and for most word-overlap metrics, a minimum of 4 references are needed in order to achieve reliable results (Finch et al., 2004). Furthermore, these metrics tend to perform poorly at segment level (Lavie & Agarwal, 2007; Chen & Cherry, 2014; Novikova et al., 2017a).

We present a novel approach to assessing NLG output quality without human references, focusing on segment-level (utterance-level) quality assessments. We train a recurrent neural network (RNN) to estimate the quality of an NLG output based on comparison with the source meaning representation (MR) only. This allows to efficiently assess NLG quality not only during system development, but also at runtime, e.g. for optimisation, reranking, or compensating low-quality output by rule-based fallback strategies.

To evaluate our method, we use crowdsourced human quality assessments of real system outputs from three different NLG systems on three datasets in two domains. We also show that adding fabricated data with synthesised errors to the training set increases relative performance by 21% (as measured by Pearson correlation).

In contrast to recent advances in referenceless quality estimation (QE) in other fields such as machine translation (MT) (Bojar et al., 2016) or grammatical error correction (Napoles et al., 2016), NLG QE is more challenging because (1) diverse realisations of a single MR are often acceptable (as the MR is typically a limited formal language); (2) human perception of NLG quality is highly variable, e.g. (Dethlefs et al., 2014); (3) NLG datasets are costly to obtain and thus small in size. Despite these difficulties, we achieve promising results – correlations with human judgments achieved by our system stay in a somewhat lower range than those achieved e.g. by state-of-the-art MT QE systems (Bojar et al., 2016), but they significantly outperform word-overlap metrics. To our knowledge, this is the first work in trainable NLG QE without references.

2. Our Model
We use a simple RNN model based on Gated Recurrent Units (GRUs) (Cho et al., 2014), composed of one GRU count. Assessing the appropriateness of responses in context is beyond the scope of this paper, see e.g. (Liu et al., 2016; Lowe et al., 2017; Cercas Curry et al., 2017).
The model assumes both the source MR and the system output to be sequences of tokens \( x^M = \{x^M_1, \ldots, x^M_m\} \), \( x^S = \{x^S_1, \ldots, x^S_n\} \), where each token is represented by its embedding (Bengio et al., 2003). The GRU encoders encode these sequences left-to-right into sequences of hidden states \( \{h^M_t\}_{t=1}^m, \{h^S_t\}_{t=1}^n \):

\[
\begin{align*}
    h^M_t &= \text{gru}(x^M_t, h^M_{t-1}) \\
    h^S_t &= \text{gru}(x^S_t, h^S_{t-1})
\end{align*}
\]

The final hidden states \( h^M_n \) and \( h^S_n \) are then fed to a set of fully connected \( \tanh \) layers \( z_i, i \in \{0 \ldots k\} \):

\[
\begin{align*}
    z_0 &= \tanh(W_0(h^M_n \circ h^S_n)) \\
    z_i &= \tanh(W_k z_{i-1})
\end{align*}
\]

The final prediction is given by a linear layer:

\[
\hat{y} = W_{k+1} z_k
\]

In (3–5), \( W_0 \ldots W_k \) stand for square weight matrices, \( W_{k+1} \) is a weight vector.

The network is trained in a supervised setting by minimising mean square error against human-assigned quality scores on the training data (see Section 3). Embedding vectors are initialised randomly and learned during training; each token found in the training data is given an embedding dictionary entry. Dropout (Hinton et al., 2012) is applied on the inputs to the GRU encoders for regularisation.

The floating point values predicted by the network are rounded to the precision and clipped to the range found in the training data.\(^3\)

### 2.1. Model Variants

We also experimented with several variants of the model which performed similarly or worse than those presented in Section 4. We list them here for completeness:

- replacing GRU cells with LSTM (Hochreiter & Schmidhuber, 1997),
- using word embeddings pretrained by the word2vec tool (Mikolov et al., 2013) on Google News data,\(^4\)
- using a set of independent binary classifiers, each predicting one of the individual target quality levels (see Section 3.1),
- using an ordered set of binary classifiers trained to predict 1 for NLG outputs above a specified quality level, 0 below it,\(^5\)
- pretraining the network using a different task (classifying MRs or predicting next word in the sentence).

### 3. Experimental Setup

In the following, we describe the data we use to evaluate our system, our method for data augmentation, detailed parameters of our model, and evaluation metrics.

#### 3.1. Dataset

Using the CrowdFlower crowdsourcing platform,\(^6\) we collected a dataset of human rankings for outputs of three recent data-driven NLG systems as provided to us by the systems’ authors; see (Novikova et al., 2017a) for more details. The following systems are included in our set:

- LOLS (Lampouras & Vlachos, 2016), which is based on imitation learning,
- RNNLG (Wen et al., 2015), a RNN-based system,
- TGen (Dušek & Jurčíček, 2015), a system using perceptron-guided incremental tree generation.

Their outputs are on the test parts of the following datasets (see Table 1):

- BAGEL (Mairesse et al., 2010) – 404 short text segments (1–2 sentences) informing about restaurants,
- SFRest (Wen et al., 2015) – ca. 5,000 segments from the restaurant information domain (including questions, confirmations, greetings, etc.),
- SFHot (Wen et al., 2015) – a set from the hotel domain similar in size and contents to SFRest.

<table>
<thead>
<tr>
<th>System</th>
<th>Data →</th>
<th>BAGEL</th>
<th>SFRest</th>
<th>SFHot</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOLS</td>
<td>202</td>
<td>581</td>
<td>398</td>
<td>1,181</td>
<td></td>
</tr>
<tr>
<td>RNNLG</td>
<td>-</td>
<td>600</td>
<td>477</td>
<td>1,077</td>
<td></td>
</tr>
<tr>
<td>TGen</td>
<td>202</td>
<td>-</td>
<td>-</td>
<td>202</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>404</td>
<td>1,181</td>
<td>875</td>
<td>2,460</td>
<td></td>
</tr>
</tbody>
</table>

\(^2\)The meaning of this number depends entirely on the training data. In our case, we use a 1–6 Likert scale assessment (see Section 3.1), but we could, for instance, use the same network to predict the required number of post-edits, as commonly done in MT (see Section 5).

\(^3\)We use a precision of 0.5 and the 1–6 range (see Section 3.1).

\(^4\)We used the model available at https://code.google.com/archive/p/word2vec/.

\(^5\)The predicted value was interpolated from classifiers’ predictions of the positive class probability.

\(^6\)http://www.crowdflower.com
Referenceless Quality Estimation for Natural Language Generation

<table>
<thead>
<tr>
<th>Instance</th>
<th>H</th>
<th>B</th>
<th>M</th>
<th>R</th>
<th>C</th>
<th>S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR</td>
<td>inform(name='la ciccia',area='bernal heights',price_range=moderate)</td>
<td>5.5</td>
<td>1</td>
<td>3</td>
<td>3.5</td>
<td>2</td>
</tr>
<tr>
<td>Ref</td>
<td>la ciccia is a moderate price restaurant in bernal heights</td>
<td>0.000</td>
<td>0.371</td>
<td>0.542</td>
<td>2.117</td>
<td></td>
</tr>
<tr>
<td>Out</td>
<td>la ciccia, is in the bernal heights area with a moderate price range.</td>
<td>2</td>
<td>4.5</td>
<td>3</td>
<td>5.5</td>
<td>2</td>
</tr>
<tr>
<td>MR</td>
<td>inform(name='intercontinental san francisco',price_range='pricey')</td>
<td>2</td>
<td>4.5</td>
<td>3</td>
<td>5.5</td>
<td>2</td>
</tr>
<tr>
<td>Ref</td>
<td>sure, the intercontinental san francisco is in the pricey range.</td>
<td>0.707</td>
<td>0.433</td>
<td>0.875</td>
<td>2.318</td>
<td></td>
</tr>
<tr>
<td>Out</td>
<td>the intercontinental san francisco is in the pricey price range.</td>
<td>2</td>
<td>4.5</td>
<td>3</td>
<td>5.5</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 2. Examples from our dataset. Ref = human reference (one selected), Out = system output to be rated. H = median human rating, B = BLEU, M = METEOR, R = ROUGE, C = CIDEr, S4 = rating given by our system in the S4 configuration. The metrics are shown with normalised and rounded values (see Section 3.5) on top and the original, raw values underneath (in italics). The top example is rated low by all metrics, our system is more accurate. The bottom one is rated low by humans but high by some metrics and our system.

Table 2. Training data size comparison for different data augmentation procedures (varies slightly in different cross-validation folds due to rounding).

3.3. Synthesising Additional Training Data

Following prior works in grammatical error correction (Rozovskaya et al., 2012; Felice & Yuan, 2014; Xie et al., 2016), we synthesise additional training instances by introducing artificial errors: Given a training instance (source MR, system output, and human rating), we generate a number of errors in the system output and lower the human rating accordingly. We use a set of basic heuristics mimicking some of the observed system behaviour to introduce errors into the system outputs:

1. removing a word,
2. duplicating a word at its current position,
3. duplicating a word at a random position,
4. adding a random word from a dictionary learned from all training system outputs,
5. replacing a word with a random word from the dictionary.

We lower the original Likert scale rating of the instance by 1 for each generated error. If the original rating was 5.5 or 6, the rating is lowered to 4 with the first introduced error and by 1 for each additional error.

Note that only values matching the source MR are delexicalised in the system outputs – if the outputs contain values not present in the source MR, they are kept intact in the model input.

To ensure that the errors truly are detrimental to the system output quality, our rules prioritise content words, i.e. they do not change articles or punctuation if other words are present. The rules never remove the last word left in the system output.
We also experiment with using additional natural language sentences where human ratings are not available – we use human-authored references from the original training datasets and assume that these would receive the maximum rating of 6. We introduce artificial errors into the human references in exactly the same way as with training system outputs.

3.4. Model Training Parameters

We set the network parameters based on several experiments performed on the development set of one of the cross-validation folds (see Section 3.1).

We train the network for 500 passes over the training data, checking Pearson and Spearman correlations on the validation set after each pass (with equal importance). We keep the configuration that yielded the best correlations overall. For setups using synthetic training data (see Section 3.3), we first perform 20 passes over all data including synthetic, keeping the best parameters, and then proceed with 500 passes over the original data. To compensate for the effects of random network initialisation, all our results are averaged over five runs with different initial random seeds following Wen et al. (2015).

3.5. Evaluation Measures

Following practices from MT Quality estimation (Bojar et al., 2016), we use Pearson’s correlation of the predicted output quality with median human ratings as our primary evaluation metric. Mean absolute error (MAE), root mean squared error (RMSE), and Spearman’s rank correlation are used as additional metrics.

We compare our results to some of the common word-overlap metrics – BLEU (Papineni et al., 2002), METEOR (Lavie & Agarwal, 2007), ROUGE-L (Lin, 2004), and CIDEr (Vedantam et al., 2015) – normalised into the 1–6 range of the predicted human ratings and further rounded to 0.5 steps. In addition, we also show the MAE/RMSE values for a trivial constant baseline that always predicts the overall average human rating (4.5).

4. Results

We test the following configurations that only differ in the amount and nature of synthetic data used (see Section 3.3 and Table 2):

- S1 Base system variant, with no synthetic data.
- S2 Adding synthetic data – introducing artificial errors into system outputs from the training portion of our dataset (no additional human references are used).
- S3 Same as previous, but with additional human references from the training portion of our dataset (including artificial errors; see Section 3.3).
- S4 As previous, but with additional human references from the training parts of the respective source NLG datasets (including artificial errors), i.e. references on which the original NLG systems were trained.
- S5 As previous, but also includes additional human references from the test portion of our dataset (including artificial errors).
- S6 As previous, but also includes development parts of the source NLG datasets (including artificial errors).

Synthetic data are never created from system outputs in the test part of our dataset. Note that S1 and S2 only use the original system outputs and their ratings, with no additional human references. S3 and S4 use additional human references (i.e. more in-domain data), but do not use human references for the instances on which the system is tested. S5 and S6 also use human references for test MRs, even if not directly, and are thus not strictly referenceless.

4.1. Results using the whole dataset

The correlations and error values we obtained over the whole data in a cross-validation setup are shown in Table 3. The correlations only stay moderate for all system variants. On the other hand, we can see that even the base setup (S1) trained using less than 2,000 examples performs better than all the word-overlap metrics in terms of all evaluation measures. Improvements in both Pearson and Spearman correlations are significant according to the Williams test (Williams, 1959; Klicickaya et al., 2017) (p < 0.01). When comparing the base setup against the constant baseline, MAE is lower but RMSE is slightly higher, which suggests that our system does better on average but is prone to occasional large errors.

The results also show that the performance can be improved considerably by adding synthetic data, especially after more than tripling the training data in S4 (Pearson correlation improvements are statistically significant in terms...
Table 3. Results using cross-validation over the whole dataset. Setups marked with “*” use human references for the test instances. All setups S1–S6 produce significantly better correlations than all metrics \((p < 0.01)\). Significant improvements in correlation \((p < 0.05)\) over S1 are marked in bold.

<table>
<thead>
<tr>
<th>Setup</th>
<th>Pearson</th>
<th>Spearman</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>~</td>
<td>~</td>
<td>1.013</td>
<td>1.233</td>
</tr>
<tr>
<td>BLEU*</td>
<td>0.074</td>
<td>0.061</td>
<td>2.264</td>
<td>2.731</td>
</tr>
<tr>
<td>METEOR*</td>
<td>0.095</td>
<td>0.099</td>
<td>1.820</td>
<td>2.129</td>
</tr>
<tr>
<td>ROUGE-L*</td>
<td>0.079</td>
<td>0.072</td>
<td>1.312</td>
<td>1.674</td>
</tr>
<tr>
<td>CIDEr*</td>
<td>0.061</td>
<td>0.058</td>
<td>2.606</td>
<td>2.935</td>
</tr>
<tr>
<td>S1: Base system</td>
<td>0.273</td>
<td>0.260</td>
<td>0.948</td>
<td>1.258</td>
</tr>
<tr>
<td>S2: + errors generated in training system outputs</td>
<td>0.283</td>
<td>0.268</td>
<td>0.948</td>
<td>1.273</td>
</tr>
<tr>
<td>S3: + training references, with generated errors</td>
<td>0.278</td>
<td>0.261</td>
<td>0.930</td>
<td>1.257</td>
</tr>
<tr>
<td>S4: + systems training data, with generated errors</td>
<td>0.330</td>
<td>0.274</td>
<td>0.914</td>
<td>1.226</td>
</tr>
<tr>
<td>S5: + test references, with generated errors*</td>
<td>0.331</td>
<td>0.265</td>
<td>0.937</td>
<td>1.245</td>
</tr>
<tr>
<td>S6: + complete datasets, with generated errors*</td>
<td>0.354</td>
<td>0.287</td>
<td>0.909</td>
<td>1.208</td>
</tr>
</tbody>
</table>

of the Williams test, \(p < 0.01\). Using additional human references for the test data seems to be helping further in S6 (the difference in Pearson correlation is statistically significant, \(p < 0.01\)). The additional references apparently provide more information even though the SFHot and SFRest datasets have similar MRs (identical when delexicalised) in training and test data (Lampouras & Vlachos, 2016).\(^{14}\) The setups using larger synthetic data further improve MAE and RMSE: S4 and S6 increase the margin against the constant baseline up to ca. 0.1 in terms of MAE, and both are able to surpass the constant baseline in terms of RMSE.

4.2. Cross-domain and Cross-System Training

Next, we test how well our approach generalises to new systems and datasets and how much in-set data (same domain/system) is needed to obtain reasonable results. We use the SFHot data as our test domain and LOLS as our test system, and we treat the rest as out-of-set. We test three different configurations:

C1 Training exclusively using a small amount of in-set data (200 instances, 100 reserved for validation), testing on the rest of the in-set.

C2 Training and validating exclusively on out-of-set data, testing on the same part of the in-set as in C1 and C3.

C3 Training on the out-of-set data with a small amount of in-set data (200 instances, 100 reserved for validation), testing on the rest of the in-set.

\(^{14}\)Note that unlike in NLG systems training on SFHot and SFRest, the identical MRs in training and test data do not allow our system to only memorize the training data as the NLG outputs to be rated are distinct. However, the situation is not 100% referenceless as the system may have been exposed to other NLG outputs for the same MR. Our preliminary experiments suggest that our systems can also handle lexicalised data well, without any modification (Pearson correlation 0.264–0.359 for S1–S6).

The results are shown in Tables 4 and 5, respectively.

The correlations of C2 suggest that while our network can generalise across systems to some extent (if data fabrication is used), it does not generalise well across domains without using in-domain training data. C1 and C3 configuration results demonstrate that even small amounts of in-set data help noticeably. However, if in-set data is used, additional out-of-set data does not improve the results in most cases (C3 is mostly not significantly better than the corresponding C1).

Except for a few cross-system C2 configurations with low amounts of synthetic data, all systems perform better than word-overlap metrics. However, most setups are not able to improve over the constant baseline in terms of MAE and RMSE.

5. Related Work

This work is the first NLG QE system to our knowledge; the most related work in NLG is probably the system by Dethlefs et al. (2014), which reranks NLG outputs by estimating their properties (such as colloquialism or politeness) using various regression models. However, our work is also related to QE research in other areas, such as MT (Specia et al., 2010), dialogue systems (Lowe et al., 2017) or grammatical error correction (Napoles et al., 2016). QE is especially well researched for MT, where regular QE shared tasks are organised (Callison-Burch et al., 2012; Bojar et al., 2013; 2014; 2015; 2016).

Many of the past MT QE systems participating in the shared tasks are based on Support Vector Regression (Specia et al., 2015; Bojar et al., 2014; 2015). Only in the past year, NN-based solutions started to emerge. Patel & M (2016) present a system based on RNN language models, which focuses on predicting MT quality on the word level.
In an evaluation spanning outputs of three different NLG systems and three datasets, our system significantly outperformed four commonly used reference-based metrics. It also improved over a constant baseline, which always predicts the mean human rating, in terms of MAE and RMSE. The smaller RMSE improvement suggests that our system is prone to occasional large errors. We have shown that generating additional training data, e.g., by using NLG training datasets and synthetic errors, significantly improves the system performance. While our system can generalise to unseen NLG systems in the same domain to some extent, its cross-domain generalisation capability is poor. However, very small amounts of in-domain/in-system data improve performance notably.

In future work, we will explore improvements to our error synthesising methods as well as changes to our network architecture (bidirectional RNNs or convolutional NNs). We also plan to focus on relative ranking of different NLG outputs for the same source MR or predicting the number of post-edits required. We intend to use data collected within the ongoing E2E NLG Challenge (Novikova et al., 2017b), which promises greater diversity than current datasets.

### Table 4. Cross-domain evaluation results. Setups marked with "*" use human references of test instances. All setups produce significantly better correlations than all metrics (p < 0.01). Significant improvements in correlation (p < 0.05) over the corresponding S1 are marked in bold, significant improvements over the corresponding C1 are underlined.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pear</td>
<td>Spea</td>
<td>MAE</td>
</tr>
<tr>
<td>Constant</td>
<td>-</td>
<td>-</td>
<td>1.060</td>
</tr>
<tr>
<td>BLEU*</td>
<td>0.079</td>
<td>0.043</td>
<td>2.514</td>
</tr>
<tr>
<td>METEOR*</td>
<td>0.141</td>
<td>0.122</td>
<td>1.929</td>
</tr>
<tr>
<td>ROUGE-L*</td>
<td>0.064</td>
<td>0.048</td>
<td>1.449</td>
</tr>
<tr>
<td>CIDEr*</td>
<td>0.127</td>
<td>0.106</td>
<td>2.801</td>
</tr>
<tr>
<td>S1</td>
<td>0.219</td>
<td>0.218</td>
<td>0.983</td>
</tr>
<tr>
<td>S2</td>
<td>0.219</td>
<td>0.218</td>
<td>0.983</td>
</tr>
<tr>
<td>S3</td>
<td>0.219</td>
<td>0.218</td>
<td>0.983</td>
</tr>
<tr>
<td>S4</td>
<td>0.219</td>
<td>0.218</td>
<td>0.983</td>
</tr>
<tr>
<td>S5</td>
<td>0.219</td>
<td>0.218</td>
<td>0.983</td>
</tr>
<tr>
<td>S6</td>
<td>0.219</td>
<td>0.218</td>
<td>0.983</td>
</tr>
</tbody>
</table>

### Table 5. Cross-system evaluation results. Setups marked with "*" use human references of test instances. Setups that do not produce significantly better correlations than all metrics (p < 0.01) are marked in italics. Significant improvements in correlation (p < 0.05) over the corresponding S1 are marked in bold, significant improvements over the corresponding C1 are underlined.

<table>
<thead>
<tr>
<th></th>
<th>C1: small in-system data only</th>
<th>C2: out-of-system data only</th>
<th>C3: out-of-sys. + small in-sys.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pear</td>
<td>Spea</td>
<td>MAE</td>
</tr>
<tr>
<td>Constant</td>
<td>-</td>
<td>-</td>
<td>1.060</td>
</tr>
<tr>
<td>BLEU*</td>
<td>0.079</td>
<td>0.043</td>
<td>2.514</td>
</tr>
<tr>
<td>METEOR*</td>
<td>0.141</td>
<td>0.122</td>
<td>1.929</td>
</tr>
<tr>
<td>ROUGE-L*</td>
<td>0.064</td>
<td>0.048</td>
<td>1.449</td>
</tr>
<tr>
<td>CIDEr*</td>
<td>0.127</td>
<td>0.106</td>
<td>2.801</td>
</tr>
<tr>
<td>S1</td>
<td>0.347</td>
<td>0.334</td>
<td>1.084</td>
</tr>
<tr>
<td>S2</td>
<td>0.358</td>
<td>0.345</td>
<td>1.007</td>
</tr>
<tr>
<td>S3</td>
<td>0.302</td>
<td>0.290</td>
<td>0.981</td>
</tr>
<tr>
<td>S4</td>
<td>0.307</td>
<td>0.296</td>
<td>1.059</td>
</tr>
<tr>
<td>S5</td>
<td>0.307</td>
<td>0.296</td>
<td>1.059</td>
</tr>
<tr>
<td>S6</td>
<td>0.307</td>
<td>0.296</td>
<td>1.059</td>
</tr>
</tbody>
</table>

Kim & Lee (2016) estimate segment-level MT output quality using a bidirectional RNN over both source and output sentences combined with a logistic prediction layer. They pretrain their RNN on large MT training data. Last year’s MT QE shared task systems achieve Pearson correlations between 0.4–0.5, which is slightly higher than our best results. However, the results are not directly comparable: First, we predict a Likert-scale assessment instead of the number of required post-edits. Second, NLG datasets are considerably smaller than corpora available for MT. Third, we believe that QE for NLG is harder due to the reasons outlined in Section 1.

6. Conclusions and Future Work

We presented the first system for referenceless quality estimation of natural language generation outputs. All code and data used here is available online at:

https://github.com/tuetschek/ratpred

In an evaluation spanning outputs of three different NLG
Acknowledgements

The authors would like to thank Lucia Specia and the two anonymous reviewers for their helpful comments. This research received funding from the EPSRC projects DILiGENt (EP/M005429/1) and MaDrIgAL (EP/N017536/1). The Titan Xp used for this research was donated by the NVIDIA Corporation.

References


Xie, Ziang, Avati, Anand, Arivazhagan, Naveen, Jurafsky, Dan, and Ng, Andrew Y. Neural Language Correction with Character-Based Attention. *arXiv:1603.09727 [cs]*, March 2016.