Identifying and Controlling Important Neurons in Neural Machine Translation

Anonymous Author(s)
Affiliation
Address
email

Abstract
Neural machine translation (NMT) models learn representations containing substantial linguistic information. However, it is not clear if such information is fully distributed or if some of it can be attributed to individual neurons. We develop unsupervised methods for discovering important neurons in NMT models. Our methods rely on the intuition that different models learn similar properties, and do not require any costly external supervision. We show experimentally that translation quality depends on the discovered neurons, and find that many of them capture common linguistic phenomena. Finally, we show how to control NMT translations in predictable ways, by modifying activations of individual neurons.

1 Introduction
Neural machine translation (NMT) systems achieve state-of-the-art results by learning from large amounts of example translations, typically without additional linguistic information. Recent studies have shown that representations learned by NMT models contain a non-trivial amount of linguistic information on multiple levels: morphological (Belinkov et al., 2017), syntactic (Shi et al., 2016b), and semantic (Hill et al., 2017). These studies use trained NMT models to generate feature representations for words, and use these representations to predict certain linguistic properties. This approach has two main limitations. First, it targets the whole vector representation and fails to analyze individual dimensions in the vector space. In contrast, previous work found meaningful individual neurons in computer vision (Zeiler & Fergus, 2014; Zhou et al., 2016; Bau et al., 2017, among others) and in a few NLP tasks (Karpathy et al., 2015; Radford et al., 2017; Qian et al., 2016a). Second, these methods require external supervision in the form of linguistic annotations. They are therefore limited by available annotated data and tools.

In this work, we make initial progress towards addressing these limitations by developing unsupervised methods for analyzing the contribution of individual neurons to NMT models. We aim to answer the following questions:
- How important are individual neurons for obtaining high-quality translations?
- Do individual neurons in NMT models contain interpretable linguistic information?
- Can we control MT output by intervening in the representation at the individual neuron level?

To answer these questions, we develop several unsupervised methods for ranking neurons according to their importance to an NMT model. Inspired by work in machine vision (Li et al., 2016b), we hypothesize that different NMT models learn similar properties, and therefore similar important neurons should emerge in different models. To test this hypothesis, we map neurons between pairs of trained NMT models using several methods: correlation analysis, regression analysis, and SVCCA, a recent method combining singular vectors and canonical correlation analysis (Raghu et al., 2017). Our mappings yield lists of candidate neurons containing shared information across models. We then evaluate whether these neurons carry important information to the NMT model by masking their activations during testing. We find that highly-shared neurons impact translation quality much more than unshared neurons, affirming our hypothesis that shared information matters.

Given the list of important neurons, we then investigate what linguistic properties they capture, both qualitatively by visualizing neuron activations and quantitatively by performing supervised classification experiments. We were able to identify neurons corresponding to several linguistic phenomena, including morphological and syntactic properties.

Finally, we test whether intervening in the representation at the individual neuron level can help control the translation. We demonstrate the ability to control NMT translations on three linguistic properties—tense, number, and gender—to varying degrees of success. This sets the ground for controlling NMT in desirable ways, potentially reducing system bias to properties like gender.

Our work indicates that not all information is distributed in NMT models, and that many human-interpretable grammatical and structural properties are captured by individual neurons. Moreover, modifying the activations of individual neurons allows controlling the translation output according to specified linguistic properties. The methods we develop here are task-independent and can be used for analyzing neural networks in other tasks. More broadly, our work contributes to the localist/distributed debate in neural cognitive science (Gayler & Levy, 2011) by investigating the important case of neural machine translation.

2 Related Work

Much recent work has been concerned with analyzing neural representations of linguistic units, such as word embeddings (Köh, 2015; Qian et al., 2016b), sentence embeddings (Adi et al., 2016; Ganesh et al., 2017; Brunner et al., 2018), and NMT representations at different linguistic levels: morphological (Belinkov et al., 2017), syntactic (Shi et al., 2016b), and semantic (Hill et al., 2017).

These studies follow a common methodology of evaluating learned representations on external supervision by training classifiers or measuring other kinds of correlations. Thus they are limited to the available supervised annotation. In addition, these studies also do not typically consider individual dimensions. In contrast, we propose intrinsic unsupervised methods for detecting important neurons based on correlations between independently trained models. A similar approach was used to analyze vision networks (Li et al., 2016b), but to the best of our knowledge these ideas were not used to study NMT or other NLP models before.

In computer vision, individual neurons were shown to capture meaningful information Zeiler & Fergus (2014); Zhou et al. (2016); Bau et al. (2017). Even though some doubts were cast on the importance of individual units (Morcos et al., 2018), recent work stressed their contribution to predicting specific object classes via masking experiments similar to ours (Zhou et al., 2018). A few studies analyzed individual neurons in NLP. For instance, neural language models learn specific neurons that activate on brackets (Karpathy et al., 2015), sentiment (Radford et al., 2017), and length (Qian et al., 2016a). Length-specific neurons were also found in NMT (Shi et al., 2016a), but generally not much work has been devoted to analyzing individual neurons in NMT. We aim to address this gap.

3 Methodology

Much recent work on analyzing NMT relies on supervised learning, where NMT representations are used as features for predicting linguistic annotations (see Section 2). However, such annotations may not be available, or constrain the analysis to a particular scheme.

Instead, we propose to use different kinds of correlations between neurons from different models as a measure of their importance. Suppose we have $M$ such models and let $h_i^m[i]$ denote the activation of the $i$-th neuron in the encoder of the $m$-th model for the $t$-th word. These may be models from different training epochs, trained with different random initializations or datasets, or even different architectures—all realistic scenarios that researchers often experiment with. Let $x_i^m$ denote a random variable corresponding to the $i$-th neuron in the $m$-th model. $x_i^m$ maps words to their neuron activations: $x_i^m : t \mapsto h_i^m[t]$. Similarly, let $x^m$ denote a random vector corresponding to the activations of all neurons in the $m$-th model: $x^m : t \mapsto h_i^m$.

We consider four methods for ranking neurons, based on correlations between pairs of models. Our hypothesis is that different NMT models learn similar properties, and therefore similar important neurons emerge in different models, akin to neural vision models (Li et al., 2016b). Our methods capture different levels of localization/distributivity, as described next. See Figure 1 for illustration.

---

1 We only consider neurons from the top layer, although the approach can also be applied to other layers.
Figure 1: An illustration of the correlation methods methods, showing how to compute the score for one neuron using each of the methods. Here the number of models is $M = 3$.

3.1 Unsupervised correlation Methods

**Maximum correlation** The maximum correlation ($\text{MaxCorr}$) of neuron $x_i^m$ looks for the highest correlation with any neuron in all other models:

$$\text{MaxCorr}(x_i^m) = \max_{j, m' \neq m} |\rho(x_i^m, x_j^{m'})|$$  \hspace{1cm} (1)

where $\rho(x, y)$ is the Pearson correlation coefficient between $x$ and $y$. We then rank the neurons in model $m$ according to their MaxCorr score. We repeat this procedure for every model $m$. This score looks for neurons that capture properties that emerge strongly in two separate models.

**Minimum correlation** The minimum correlation ($\text{MinCorr}$) of neuron $x_i^m$ looks for the neurons most correlated with $x_i^m$ in each of the other models, but selects the one with the lowest correlation:

$$\text{MinCorr}(x_i^m) = \min_{m' \neq m} \max_j |\rho(x_i^m, x_j^{m'})|$$  \hspace{1cm} (2)

Neurons in model $m$ are ranked according to their MinCorr score. This tries to find neurons that are well correlated with many other models, even if they are not the overall most correlated ones.

**Regression ranking** We perform linear regression ($\text{LinReg}$) from the full representation of another model $x_i^{m'}$ to the neuron $x_i^m$. Then we rank neurons by the regression mean squared error. This attempts to find neurons whose information might be distributed in other models.

**SVCCA** Singular vector canonical correlation analysis ($\text{SVCCA}$) is a recent method for analyzing neural networks (Raghu et al., 2017). In our implementation, we perform PCA on each model’s representations $x^m$ and take enough dimensions to account for 99% of the variance. For each pair of models, we obtain the canonically correlated basis, and rank the basis directions by their CCA coefficients. This attempts to capture information that may be distributed in less dimensions than the whole representation. In this case we get a ranking of directions, rather than individual neurons.

3.2 Verifying Detected Neurons

We want to verify that neurons ranked highly by the unsupervised methods are indeed important for the NMT models. We consider quantitative and qualitative techniques for verifying their importance.

**Erasing Neurons** We test importance of neurons by erasing some of them during translation. Erasure is a useful technique for analyzing neural networks (Li et al., 2016a). Given a ranked list of neurons $\pi$, where $\pi(i)$ is the rank of neuron $x_i$, we zero-out increasingly more neurons according to the ranking $\pi$, starting from either the top or the bottom of the list. Our hypothesis is that erasing neurons from the top would hurt translation performance more than erasing from the bottom.

Concretely, we first run the entire encoder as usual, then zero out specific neurons from all source hidden states $\{h_1, \ldots, h_n\}$ before running the decoder. For MaxCorr, MinCorr, and LinReg, we zero out individual neurons. To erase $k$ directions found by SVCCA, we instead project the embedding $E$ (corresponding to all activations of a given model over a dataset) onto the space spanned by the non-erased directions: $E' = E(C(C^T C)^{-1}C^T)$, where $C$ is the CCA projection matrix with the first or last $k$ columns removed. This corresponds to erasing from the top or bottom.

**Supervised Verification** While our focus is on unsupervised methods for finding important neurons, we also utilize supervision to verify our results. Since training a supervised classifier on every neuron is costly, we instead report simple metrics that can be easily computed. Specifically, we
Figure 2: Erasing neurons (or SVCCA directions) from the top and bottom of the list of most important neurons (directions) ranked by different unsupervised methods, in an English-Spanish model.

4 Experimental Setup

Data We use the United Nations (UN) parallel corpus (Ziemski et al., 2016) for all experiments. We train models from English to 5 languages: Arabic, Chinese, French, Russian, and Spanish, as well as an English-English auto-encoder. For each target language, we train 3 models on different parts of the training set, each with 500K sentences. In total, we have 18 models. This setting allows us to compare models trained on the same language pairs but different training data, as well as models trained on different language pairs. We evaluate on the official test set.

MT training We train 500 dimensional 2-layer LSTM encoder-decoder models with attention Bahdanau et al. (2014). In order to study both word and sub-word properties, we use a word representation based on a character convolutional neural network (charCNN) as input to both encoder and decoder, which was shown to learn morphology in language modeling and NMT (Kim et al., 2015; Belinkov et al., 2017). ² While we focus here on recurrent NMT, our approach can be applied to other models like the Transformer (Vaswani et al., 2017), which we leave for future work.

5 Results

5.1 Erasure Experiments

Figure 2 shows erasure results using the methods from Section 3.1, on an English-Spanish model. For all four methods, erasing from the top hurts performance much more than erasing from the bottom. This confirms our hypothesis that neurons ranked higher by our methods have a larger impact on translation quality. Comparing erasure with different rankings, we find similar patterns with MaxCorr, MinCorr, and LinReg: erasing the top ranked 10% (50 neurons) degrades BLEU by 15-20 points, while erasing the bottom 10% neurons only hurts by 2-3 points. In contrast, erasing SVCCA directions results in rapid degradation – 15 BLEU point drop when erasing 1% (5) of the top directions, and poor performance when erasing 10% (50). This indicates that top SVCCA directions capture very important information in the model. We analyze these top neurons and directions in the next section, finding that top SVCCA directions focus mostly on identifying specific words.

Figure 3 shows the results of MaxCorr when erasing neurons from top and bottom, using models trained on three language pairs. In all cases, erasing from the top hurts performance more than erasing from the bottom. We found similar trends with other language pairs and ranking methods.

5.2 Evaluating Top Neurons

What kind of information is captured by the neurons ranked highly by each of our ranking methods? Previous work found specific neurons in NMT that capture position of words in the sentence (Shi ²We used this representation rather than BPE sub-word units (Sennrich et al., 2016) to facilitate interpretability with respect to specific words. In the experiments, we report word-based results unless noted otherwise.}
et al., 2016a). Do our methods capture similar properties? Indeed, we found that many of the top neurons capture position. For instance, Table 1 shows the top 10 ranked neurons from an English-Spanish model according to each of the methods. The table shows the percent of variance in neuron activation that is eliminated by conditioning on position in the sentence, calculated over the test set. Similarly, it shows the percent of explained variance by conditioning on the current token identity. We observe an interesting difference between the ranking methods. *MaxCorr* and especially *MinCorr*, which are both computed by using multiple neurons, tend to find information determined by the identity of the current token. *LinReg* and (especially) *SVCCA* tend to find position information. This suggests that information about the current token is often distributed in multiple neurons, which can be explained by the fact that tokens carry multiple kinds of linguistic information. In contrast, position is a fairly simple property that the NMT encoder can represent in a small number of neurons.

### 5.3 Linguistically Interpretable Neurons

Neurons that activate on specific tokens or capture position in the sentence are important, as shown in the previous section. But they are less interesting from the perspective of capturing language information. In this section, we investigate several linguistic properties by measuring predictive capacity and visualizing neuron activations.

#### Parentheses

Table 2 shows top neurons from each model for predicting that tokens are inside/outside of parentheses, quotes, or brackets, estimated by a GMM model. Often, the parentheses neuron is unique (low scores for the 2nd best neuron), suggesting that this property tends to be relatively localized. Generally, neurons that detect parentheses were ranked highly in most models by the *MaxCorr* method, indicating that they capture important patterns in multiple networks.

The next figure visualizes the most predictive neuron in an English-Spanish model. It activates positively (red) inside parentheses and negatively (blue) outside. Similar neurons were found in RNN language models (Karpathy et al., 2015). Next we consider more complicated linguistic properties.

#### Tense

We annotated the test data for verb tense (with Spacy) and trained a GMM model to predict tense from neuron activations. The following figure shows activations of a top-scoring neuron (0.56 F1) from the English-Arabic model on the first 5 test sentences. It tends to activate positively (red color) on present tense (“recognizes”, “recalls”, “commemorate”) and negatively (blue color) on past...
Table 2: $F_1$ scores of the top two neurons from each network for detecting tokens inside parentheses, and the ranks of the top neuron according to our intrinsic unsupervised methods.

<table>
<thead>
<tr>
<th>Neuron</th>
<th>1st</th>
<th>2nd</th>
<th>Max</th>
<th>Min</th>
<th>Reg</th>
<th>Neuron</th>
<th>1st</th>
<th>2nd</th>
<th>Max</th>
<th>Min</th>
<th>Reg</th>
</tr>
</thead>
<tbody>
<tr>
<td>en-es-1:232</td>
<td>0.59</td>
<td>0.3</td>
<td>14</td>
<td>44</td>
<td>26</td>
<td>en-ar-3:331</td>
<td>0.59</td>
<td>0.35</td>
<td>17</td>
<td>92</td>
<td>49</td>
</tr>
<tr>
<td>en-es-2:208</td>
<td>0.72</td>
<td>0.26</td>
<td>8</td>
<td>43</td>
<td>21</td>
<td>en-ru-1:259</td>
<td>0.64</td>
<td>0.33</td>
<td>10</td>
<td>47</td>
<td>44</td>
</tr>
<tr>
<td>en-es-3:47</td>
<td>0.57</td>
<td>0.29</td>
<td>11</td>
<td>34</td>
<td>23</td>
<td>en-ru-2:23</td>
<td>0.71</td>
<td>0.26</td>
<td>10</td>
<td>72</td>
<td>31</td>
</tr>
<tr>
<td>en-fr-1:499</td>
<td>0.6</td>
<td>0.27</td>
<td>37</td>
<td>41</td>
<td>14</td>
<td>en-ru-3:214</td>
<td>0.65</td>
<td>0.32</td>
<td>25</td>
<td>67</td>
<td>114</td>
</tr>
<tr>
<td>en-fr-2:361</td>
<td>0.61</td>
<td>0.35</td>
<td>28</td>
<td>44</td>
<td>60</td>
<td>en-zh-1:49</td>
<td>0.58</td>
<td>0.44</td>
<td>5</td>
<td>85</td>
<td>63</td>
</tr>
<tr>
<td>en-fr-3:253</td>
<td>0.37</td>
<td>0.35</td>
<td>140</td>
<td>122</td>
<td>68</td>
<td>en-zh-2:159</td>
<td>0.76</td>
<td>0.38</td>
<td>5</td>
<td>47</td>
<td>37</td>
</tr>
<tr>
<td>en-ar-1:383</td>
<td>0.38</td>
<td>0.36</td>
<td>119</td>
<td>195</td>
<td>228</td>
<td>en-zh-3:467</td>
<td>0.54</td>
<td>0.32</td>
<td>5</td>
<td>59</td>
<td>47</td>
</tr>
<tr>
<td>en-ar-2:166</td>
<td>0.63</td>
<td>0.25</td>
<td>4</td>
<td>117</td>
<td>67</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Strongest correlations in all models relative to a tense neuron in an English-Arabic model.

<table>
<thead>
<tr>
<th>Arabic</th>
<th>0.66, 0.57</th>
</tr>
</thead>
<tbody>
<tr>
<td>French</td>
<td>-0.69, -0.58, -0.48</td>
</tr>
<tr>
<td>Russian</td>
<td>-0.50, -0.39, -0.29</td>
</tr>
<tr>
<td>Chinese</td>
<td>-0.51, -0.30, -0.18</td>
</tr>
<tr>
<td>English</td>
<td>-0.33, -0.19, -0.03</td>
</tr>
</tbody>
</table>

tense (“published”, “disbursed”, “held”). These results are obtained with a charCNN representation, which is sensitive to common suffixes like “-ed”, “-es”. However, this neuron also detects irregular past tense verbs like “held”, suggesting that it captures context in addition to sub-word information. The neuron also makes some mistakes by activating weakly positively on nouns ending with “s” (“videos”, “punishments”), presumably because it gets confused with the 3rd person present tense.

Table 3 shows correlations of neurons most correlated with this tense neuron, according to MaxCorr. All these neurons are highly predictive of tense: all are in the top 5 and 9 out of 15 (non-auto-encoder) neurons have the highest $F_1$ score for predicting tense. The auto-encoder English models are an exception, exhibiting much lower correlations with the English-Arabic tense neuron. This suggests that tense emerges in a “real” NMT model, but not in an auto-encoder that only learns to copy. Interestingly, English-Chinese models have somewhat lower correlated neurons with the tense neuron, possibly due to the lack of explicit tense marking in Chinese. The encoder does not need to pay as much attention to tense when generating representations for the decoder.

Other Properties. We found many more linguistic properties by visualizing top neurons ranked by our methods, especially with MaxCorr. We found neurons that activate on numbers, dates, adjectives, plural nouns, auxiliary verbs, prepositions, and more. We do not include a detailed discussion for lack of space, and instead briefly discuss noun phrase segmentation, a compositional property above the word level. We obtained noun phrase segmentation (using Spacy) and classified tokens as inside, outside, or beginning of a noun phrase (IOB scheme), and found high-scoring neurons (60-80% accuracy) in every network. Many of these neurons were ranked highly by the MaxCorr method. In contrast, other methods did not rank such neurons very highly.

We visualize the top scoring neuron (79%) from an English-Spanish model below. Notice how the neuron activates positively (red color) on the first word in the noun phrases, but negatively (blue color) on the rest of the noun phrase (e.g. “Regional” in “Regional Service Centre”). This neuron is the 9th highest ranked neuron in an English-Spanish model according to MaxCorr.
In this section, we explore a potential benefit of finding important neurons with linguistically meaningful properties: controlling the translation output. This may be important for mitigating biases in neural networks. For instance, gender stereotypes are often reflected in automatic translations, as the following motivating examples from Google Translate demonstrate.

1. a. o bir doctor  
   b. he is a doctor

2. a. o bir hemşire  
   b. she is a nurse

The Turkish sentences (1a, 2a) have no gender information—they can refer to either male or female. But the MT system is biased to think that doctors are usually men and nurses are usually women, so its generated translations (1b, 2b) represent these biases.

We conjecture that if a given neuron matters to the model, then we can control the translation in predictable ways by modifying its activations. To do this, we first encode the source sentence as usual. Before decoding, we set the activation of a particular neuron in the encoder state to a value $\alpha$, which is a function of the mean activations over a particular property (defined below). To evaluate our ability to control the translation, we design the following protocol:

1. Tag the source and target sentences in the development set with a desired property, such as gender (masculine/feminine). We use Spacy for these tags.
2. Obtain word alignments for the development set with using an alignment model trained on 2 million sentences of the UN data. We use fast_align (Dyer et al., 2013) with default settings.
3. For every neuron in the encoder, predict the target property on the word aligned to its source word activations using a supervised GMM model.$^4$
4. For every word having a desired property, modify the source activations of the top $k$ neurons found in step 3, and generate a modified translation. The modification value is defined as $\alpha = \mu_1 + \beta(\mu_1 - \mu_2)$, where $\mu_1$ and $\mu_2$ are mean activations of the property we modify from and to, respectively (e.g. modifying gender from masculine to feminine), and $\beta$ is a hyper-parameter.
5. Tag the output translation and word-align it to the source. Declare success if the source word was aligned to a target word with the desired property value (e.g. feminine).

### 6.1 Results

Figure 4 shows translation control results in an English-Spanish model. We report success rate—the percentage of cases where the word was aligned to a target word with the desired property—and the effect on BLEU scores, when varying $\alpha$. Our tense control results are the most successful, with up to 67% success rate for changing past-to-present. Modifications generally degrade BLEU, but the loss at the best success rate is not large (2 BLEU points).

Controlling other properties seems more difficult, with the best success rate for controlling number at 37%, using the 5 top number neurons. Gender is the most difficult to control, with a 21% success rate using the 5 top neurons. Modifying even more neurons did not help. We conjecture that these properties are more distributed than tense, which makes controlling them more difficult. Future work can explore more sophisticated methods for controlling multiple neurons simultaneously.

---

$^3$For more biased examples, see mashable.com/2017/11/30/google-translate-sexism.

$^4$This is different from our results in the previous section, where we predicted a source-side property, because here we seek neurons that are predictive of target-side properties to facilitate controlling the translation.
Table 4: Examples for controlling translation by modifying activations of different neurons on the italicized source words. $\alpha =$ modification value ($-\alpha$, no modification).

(a) Controlling number when translating “The interested parties” to Spanish.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Translation</th>
<th>Num</th>
<th>$\alpha$</th>
<th>Translation</th>
<th>Num</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>abiertas particulares</td>
<td>pl.</td>
<td>0.125</td>
<td>La parte interesada</td>
<td>sing.</td>
</tr>
<tr>
<td>-0.5</td>
<td>Observaciones interesadas</td>
<td>pl.</td>
<td>0.25</td>
<td>Cuestión interesada</td>
<td>sing.</td>
</tr>
<tr>
<td>-0.25, -0.125, 0</td>
<td>Las partes interesadas</td>
<td>pl.</td>
<td>0.5, 1</td>
<td>Gran útil</td>
<td>sing.</td>
</tr>
</tbody>
</table>

(b) Controlling gender when translating “The interested parties” (left) and “Questions relating to information” (right) to Spanish.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Translation</th>
<th>Gen</th>
<th>$\alpha$</th>
<th>Translation</th>
<th>Gen</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.5, -0.25</td>
<td>Los partidos interados</td>
<td>ms.</td>
<td>-1</td>
<td>Temas relativos a la información</td>
<td>ms.</td>
</tr>
<tr>
<td>0, 0.25</td>
<td>Las partes interesadas</td>
<td>fm.</td>
<td>-0.5, 0, 0.5</td>
<td>Cuestiones relativas a la información</td>
<td>fm.</td>
</tr>
</tbody>
</table>

(c) Controlling tense when translating “The committee supported the efforts of the authorities”.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>Translation</th>
<th>Tense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>$-\alpha$+10</td>
<td>past/present</td>
</tr>
<tr>
<td>French</td>
<td>$-\alpha$-20</td>
<td>past/present</td>
</tr>
<tr>
<td>Spanish</td>
<td>$-\alpha$-3/0</td>
<td>past/impf./present</td>
</tr>
<tr>
<td>Russian</td>
<td>$-\alpha$-1</td>
<td>past/present</td>
</tr>
<tr>
<td>Chinese</td>
<td>$-\alpha$-50</td>
<td>untensed/present</td>
</tr>
</tbody>
</table>

6.2 Example translations

We provide examples of controlling translation of number, gender, and tense. While these are cherry-picked examples, they illustrate that the controlling procedure can work in multiple properties and languages. We discuss language-specific patterns below.

**Number** Table 4a shows translation control results for a number neuron from an English-Spanish model, which activates negatively/positively on plural/singular nouns. The translation changes from plural to singular as we increase the modification $\alpha$. We notice that using too high $\alpha$ values yields nonsense translations, but with correct number: transitioning from the plural adjective *particulares* ("particular") to the singular adjective *útil* ("useful"), with valid translations in between.

**Gender** Table 4b shows examples of controlling gender translation for a gender neuron from the same model, which activates negatively/positively on masculine/feminine nouns. The translations change from masculine to feminine synonyms as we increase the modification $\alpha$. Generally, we found it difficult to control gender, as also suggested by the relatively low success rate.

**Tense** Table 4c shows examples of controlling tense when translating from English to five target languages. In all language pairs, we are able to change the translation from past to present by modifying the activation of the tense neurons from the previous section (Table 3). In Spanish, we find a transition from past to imperfect to present. Interestingly, in Chinese, we had to use a fairly large $\alpha$ value (in absolute terms), consistent with the fact that tense is not usually marked in Chinese.

7 Conclusion

We developed unsupervised methods for finding important neurons in NMT, and evaluated how these neurons impact translation quality. We analyzed several linguistic properties that are captured by individual neurons using quantitative prediction tasks and qualitative visualizations. We also designed a protocol for controlling translations by modifying neurons that capture desired properties.

Our analysis can be extended to other NMT components (e.g. the decoder) and architectures Gehring et al. (2017); Vaswani et al. (2017), as well as other tasks. We believe that more work should be done to analyze the spectrum of localized vs. distributed information in neural language representations. We would also like to develop more sophisticated ways to control translation output, for example by modifying representations in variational NMT architectures (Zhang et al., 2016; Su et al., 2018).
References


