Rare and novel words in SQuAD 2.0: An application of the form-context-model

Stanford CS224N Default Project

Axel Gross-Klussmann
Department of Computer Science
Stanford University
agk1982@stanford.edu

Abstract

Motivated by shortcomings of word embeddings like Glove in handling out-of-vocabulary and rare words, our study sheds light on the performance of SQuAD 2.0 models when question and answer contexts are dominated by rare and novel word occurrence. To improve model performances we employ several setups based on sub-word information. First, we extend the BiDAF baseline with character embeddings (1). Second, we utilize the form-context model with attentive mimicking (FCM+AM, see [2], [3], and [4]) to leverage and balance out both word context and word form (subword) information in order to obtain high quality embeddings for novel and rare words. Third, we build on the main idea behind the FCM+AM and implement a gating mechanism administering when to rely on word surface embeddings and when to rely on word context embeddings. Our results show that accounting for both form and context improves main metrics for SQuAD 2.0 evaluation over using pure character embeddings.

1 Introduction

My main approach consists of utilizing and improving the architecture of the form-context model (FCM) put forward by Schick and Schuetze ([3], [4], and [2]) to learn high-quality representations for rare words. In specific, I implement and refine architectures for the BiDAF based on the FCM idea in order to improve question answering in the SQuAD 2.0 task by explicitly addressing on word rarity.

The major idea of the FCM is to cleverly combine both surface-form-based (i.e. subword information) and word-representation-based approaches to get better rare word embeddings. The authors [3], [4], and [2] calibrate embeddings $v_{w,C}$ for rare words $w$ and contexts $C$ as follows:

$$v_{w,C} = \alpha \cdot A v_{w,C}^{\text{context}} + (1 - \alpha) \cdot v_{w,C}^{\text{form}}.$$

(1)

The matrix $A$ and the weightings $\alpha$ are learned parameters. The embeddings $v_{w,C}^{\text{context}}$ are weighted averages of word representation embeddings in contexts around $w$. Vectors $v_{w,C}^{\text{form}}$ are surface-form embeddings of $w$ given as averages over n-gram embeddings.

Although $\alpha$ is a scalar parameter, a hidden sigmoid layer can be used to have a 'gated' parameter $\alpha$:

$$\alpha = \sigma(w^T [v_{w,C}^{\text{form}}, v_{w,C}^{\text{context}}] + b).$$

(2)

The authors aim to improve the combination of surface-form and word representations, eq. [1] further by attentive mimicking. This approach takes up the idea of self-attention from [5] in order to train weights of the weighted average $v_{w,C}^{\text{context}}$. In specific, weights are trained such that more informative contexts for the rare words receive higher weights in the weighted average. The weight of a context, i.e. the context reliability is based on the similarity between two contexts,

$$s(C_1, C_2) = \frac{(M v_{C_1}) \cdot (M v_{C_2})^T}{\sqrt{d}}.$$

(3)
M is later learned along the way during training. The context embedding is given as a weighted average of word embeddings in the given contexts,

$$v_{\text{context}} = \sum_{C \in \mathcal{C}} \sum_{C' \in \mathcal{C}} s(C, C') v_C \rho(C).$$

(4)

where $v_C$ denotes the average of all word embeddings in a context $C$ and $\rho(C)$ is the context weight.

2 Related Work

Next to [2, 3] and [4], the FCM builds on seminal papers on word embeddings like [6] as well as on subword modeling like [7]. The use of features and self-attention utilized in the later sections is inspired by the setting of [3] and [9].

3 Approach / architecture

The foundation of my approach is the Bidirectional Attention Flow (BiDAF) neural network by [1]. I experiment with several extension to the basic setup without character embeddings:

1. I train the FCM model by [2]. This model is used to obtain word embeddings for SQuAD’s out-of-vocabulary (OOV) and rare words. No character embeddings are needed in this case.
2. In an alternative to 1. I add character embeddings to the BiDAF. Then I augment the BiDAF with an FCM inspired-gate governing the weight combination of context and surface, i.e. character, embeddings per word.
3. Given the typically stronger word embeddings for frequent words (see [2]), I experiment with external features like word frequencies based on the WestburyLab Wikipedia Corpus (WWC, [10]). Following [9] I add part of speech tags. The features are supposed to support the context vs. surface decision in the FCM style gate.
4. Finally, I adapt the self-attention idea successfully employed in [8] for SQuAD 2.0. The self-attention operates solely on the question to answer attention part of the BiDAF.

The $v_{\text{form}}$ embedding for a word $w$ in [3] is based on averages of n-grams of the word. To mimic this somewhat I implement a CNN layer for the given character embeddings. Let $x_{\text{reshaped}} \in \mathbb{R}^{e_{\text{char}} \times m_{\text{word}}}$ be the character embedding of a word of maximum length $m_{\text{word}}$ after looking up the $e_{\text{char}}$-dimensional embeddings for character indices.

Let Conv1d be a convolutional layer with kernel size 5 and $f$ output channels. Our conv. net layer is given as

$$x_{\text{conv}} = \text{Conv1D}(x_{\text{reshaped}}) \in \mathbb{R}^{f \times (m_{\text{word}} - 4)}$$

$$x_{\text{conv},\text{out}} = \text{MaxPool}(\text{ReLU}(x_{\text{conv}})) \in \mathbb{R}^{f}$$

Both the word embeddings $v_i$ and the convoluted character embedding is now projected to $\mathbb{R}^H$, where $H$ is the hidden size:

$$h_i^c = W_{\text{proj}}^c x_{\text{conv},\text{out}} \in \mathbb{R}^H$$

$$h_i^w = W_{\text{proj}}^w v_i \in \mathbb{R}^H,$$

where superscript $w$ denotes the word embedding and $c$ the character embedding. Motivated by the FCM setup we control the context-surface mixed embedding explicitly with a gate,

$$h_i' = g \odot h_i^c + (1 - g) \odot h_i^w \in \mathbb{R}^H,$$

where the gating is controlled via an element-wise sigmoid,

$$x_{\text{gate}} = \sigma(W_{\text{proj}} x_i + b_{\text{proj}}).$$

(5)

The inputs to the gate are based on exogenous features. Given the findings in [2] I hypothesize that the number of word occurrences plays a large role in the relative importance of context and surface
embeddings over one another. The raw input features \( x_i \) to the gate \( \tilde{a}_i \) are the part of speech tag for the word under consideration and the normalized number of occurrences in the WWC corpus,

\[
x_i = [\text{pos}_i, \text{nrw}_i] \in \mathbb{R}^{P+1},
\]

where the part of speech tag is a one-hot vector of size \( P \) and the normalized number of occurrences are scalar.

\[\text{[8]}\] add self-attention to the context to question (C2Q) attention of their SQuAD model. In a similar vein I use a one-directional gated recurrent unit (GRU) to encode the C2Q attention \( a_i \) of the BiDAF as follows:

\[
\hat{a}_i = \text{GRU}(\hat{a}_{i-1}, a_i) \in \mathbb{R}^H.
\]

In second step I populate a similarity matrix for the self (attention to attention) attention,

\[
S_{ij} = w^T [\hat{a}_i, \hat{a}_j, \hat{a}_i \odot \hat{a}_j]
\]

The similarity is used to get the C2Q self-attention outputs \( s_{a_i} \),

\[
\hat{s}_{a_i} = \text{softmax}(S_{i,:})
\]

\[
s_{a_i} = \sum_j \hat{s}_{a_{i,j}} \hat{a}_j.
\]

Finally I collect the traditional BiDAF attention vector and concatenate the self-attention,

\[
g^*_i = [g_i, s_{a_i}] \in \mathbb{R}^{(8+1)H}.
\]

The remaining layers are as given in the standard BiDAF setting.

4 Experiments

4.1 Data

Training the FCM model \([\text{[2]}]\) is based on the WWC corpus. As a side effect, the given code for the baseline fcm model gives word frequencies for every word occurring in the large WWC corpus. Further, \([\text{[2]}]\) introduces the WNLaMPro (WordNet Language Model Probing) dataset which contains rare words for testing language models. Given the word frequencies from WWC as well as the rare words from WNLaMPro allows me to evaluate the SQuAD system in rare word contexts and conditional on word occurrences. Other than the two outside datasets I use the SQuAD train data to train my models and the dev data to evaluate before submission to the test leaderboard.

4.2 Evaluation method

Next to the full sample dev and test set F1 and EM scores I aim to shed light on interesting subsets of the SQuAD data. To get a better grasp of the model performance with respect to infrequent words I evaluate F1 and EM on subsets of context-question pairs with rare words or, e.g., low word frequencies. To seperate contexts with rare words I use the WNLaMPro rare word column where I take words with a frequency of below 456 as rare. Then I cut the SQuAD dev data to just those contexts containing one of the rare words. The rare word subset of the dev data will be called ‘RW’ henceforth. The RW dev subset is most likely to contain OOV words.

For the second part of my model evaluation given certain word frequencies I proceed as follows:

1. I use the word frequencies from the WWC data and compute average word frequencies for every context in the SQuAD dev set.
2. I compute quartiles of the avg. context word frequencies obtained in 1.
3. The SQuAD dev data is split according to inner quartile-ranges. This means I collect contexts with average word frequencies between avg. word frequency quartiles along the splits \([0, q_1], [q_1, q_2], [q_2, q_3], [q_3, q_4]\), where \( q_i \) denotes the quartile of the context average word frequency. Ultimately this yields four groups, denoted Q1 for \([0, q_1]\), Q2 for \([q_1, q_2]\), Q3 for \([q_2, q_3]\) and Q4 for \([q_3, q_4]\).

The four context groups Q1,..,Q4 based on the dev set are hence sorted by the average word frequency. F1 and EM scores can be computed routinely on the dev subsets.
4.3 Experimental details

I first use the configuration in [2] to learn the baseline, raw, form context model proposed by the authors. This gives refined Glove word embeddings I could use as replacement of the given Glove embeddings. Second, as discussed in section 3, I take up ideas from the FCM and incorporate them in the BiDAF. In this context I try several neural network architecture modifications. Most of the runs use the starting configuration of the BiDAF. Altogether, the training time is varying, up to 16 hours. Smaller changes to the starter config. were made as follows:

- I replaced all LSTMs with GRUs to keep the model more parsimonious.
- The learning rate I tested were 0.4, 0.5 and 0.6
- I tried out dropout probabilities 0.2 and 0.3.

4.4 Results

Table [1] shows key metrics and results for the full dev set sample as well as the context groups Q1, ..., Q4 and RW defined above. The rows correspond to different specifications tested. While the first row gives the baseline, the next rows show:

1. new FCM embeddings: Rare word and OOV Glove embeddings of the BiDAF replaced by fine-tuned embeddings ([2]). Outside of the OOV and rare word embeddings, the word embeddings are still the standard Glove embeddings.
2. FCMchar: The character-augmented BiDAF with an FCM-style gating mechanism, equation 3 in section 3 handling the degree of context and surface embedding used for words. The sigmoid function of the gate is dependent on the character and word embeddings itself in this case.
3. FCMchar + feat (i): The gating of the form - context vector depends on the part of speech tag, the word frequency as well as the word and character embedding. The POS tag and the word frequency are concatenated to the gated embedding later: \( x_i = [\text{pos}_i, \text{nrw}_i, e_{i, \text{gated}}] \).
4. FCMchar + feat (ii): The gating depends just on the part of speech and WWC word frequency (see equation 6 in section 3). It explicitly does not depend on the word and character embedding. The external features are used only for the gating and NOT concatenated to the gated embedding.
5. FCMchar + feat + RNet: The self attention mechanism 7 of section 3 added to the specification FCMchar + feat (i).
6. char + feat + RNet: The self attention mechanism 7 of section 3 added to the BiDAF baseline including character embeddings and external features, but no FCM-style gating mechanism.

The following findings emerge from table 1. Contrary to intuition, the FCM-augmented BiDAF performs slightly worse than the baseline. I conjecture the reason for the poor performance is the small context piece available for each OOV token in the WWC data such that these embeddings are of no use. Notably, the raw FCM is not directly trained to the SQuAD data but just used to get OOV and rare word embeddings.

The findings with regard to the FCM-style gate for the BiDAF are inconclusive. I observe good results for dev subsets with low word frequencies (F1 (Q1), F1 (Q2)) as well as rare words (F1 (RW)) for the FCMchar models augmented by POS and word frequency features. However, in terms of strength these results are matched by the model without gating mechanism (char + feat + RNet) which has self-attention and external features. The latter finding shows that the gains from the FCM gating are small compared to refined attention mechanisms and the use of more features.

On the upside, the FCM style gating somewhat confirms results of [2] when combined with features like the POS tag and importantly, the word frequency in the WWC corpus data: The FCM configurations with external features show consistently good results in the low frequency dev set quartile (Q1) and the contexts containing rare words (RW). This is an up-front expected result. Conditional on low word frequencies or OOV words the gate should switch to the character embedding.

\(^1\)Code given at [https://github.com/timoschick/form-context-model](https://github.com/timoschick/form-context-model)
A beneficial side effect of a FCM-style gate which depends on, e.g. the word frequency as external feature, seems to be the better training performance of the model 'FCMchar+feat (ii)'. In other words, the external input features seem to improve training in early stages. Upon observing the smoothly increasing F1 and EM scores during training I increase the training time for the FCMchar+feat (ii). The corresponding model is included as last row in table §. It strongly outperforms its competitors on the dev set.

<table>
<thead>
<tr>
<th>model: BiDAF+...</th>
<th>EM(all)</th>
<th>F1(all)</th>
<th>F1(Q1)</th>
<th>F1(Q2)</th>
<th>F1(Q3)</th>
<th>F1(Q4)</th>
<th>F1(RW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>... (baseline)</td>
<td>57.7</td>
<td>61.2</td>
<td>60.7</td>
<td>60.7</td>
<td>62.4</td>
<td>61.3</td>
<td>58.7</td>
</tr>
<tr>
<td>new FCM embeddings</td>
<td>57.9</td>
<td>61.7</td>
<td>60.3</td>
<td>60.8</td>
<td>62.4</td>
<td>61.2</td>
<td>59.7</td>
</tr>
<tr>
<td>FCMchar</td>
<td>57.2</td>
<td>60.9</td>
<td>59.4</td>
<td>60.2</td>
<td>61.2</td>
<td>62.6</td>
<td>54.5</td>
</tr>
<tr>
<td>FCMchar+feat (i)</td>
<td>59.4</td>
<td>63.0</td>
<td>62.8</td>
<td>61.7</td>
<td>63.3</td>
<td>64.2</td>
<td>61.8</td>
</tr>
<tr>
<td>FCMchar+feat (ii)</td>
<td>59.0</td>
<td>62.6</td>
<td>62.2</td>
<td>62.4</td>
<td>62.2</td>
<td>63.6</td>
<td>67.9</td>
</tr>
<tr>
<td>FCMchar+feat+RNet</td>
<td>59.8</td>
<td>63.4</td>
<td>60.8</td>
<td>63.6</td>
<td>64.9</td>
<td>64.1</td>
<td>66.3</td>
</tr>
<tr>
<td>char+feat+RNet</td>
<td>60.1</td>
<td>63.5</td>
<td>62.6</td>
<td>62.4</td>
<td>64.3</td>
<td>64.8</td>
<td>57.2</td>
</tr>
<tr>
<td>FCMchar+feat (ii)*</td>
<td>60.5</td>
<td>64.3</td>
<td>63.9</td>
<td>64.1</td>
<td>63.7</td>
<td>65.5</td>
<td>62.9</td>
</tr>
</tbody>
</table>

Table 1: Dev Set analysis: EM(all) and F1(all) give scores for the full SQuAD dev examples. F1(Q1),..,F1(Q4) give the F1 score for examples corresponding to inter-quartile groups of average word frequency per context. The last row corresponds to the model of row 5 with much longer training.

Out of the several models trained I have submitted the basic FCMchar including external features with the gate fully dependent on the features and word and character embeddings to the test leaderboard. The second submission was the FCMchar with external features just used in the FCM-style gate. This is the model trained for a longer time (see the last row in table §). Finally, I submitted the gate-less but self-attention augmented CharBiDAF. Even though the FCMchar+feat (ii) model had by far the best dev set performance, it does not generalize well to the test set, indicating overfitting of the training and dev data. Still, the results from this model are the best out of the three submissions.

<table>
<thead>
<tr>
<th>model: BiDAF+...</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCMchar+feat (i)</td>
<td>57.515</td>
<td>61.473</td>
</tr>
<tr>
<td>FCMchar+feat (ii)*</td>
<td>58.09</td>
<td>61.827</td>
</tr>
<tr>
<td>char+feat+RNet</td>
<td>58.09</td>
<td>61.631</td>
</tr>
</tbody>
</table>

Table 2: Test set results based on three submissions. Best position on non-PCE leaderboard: 21st.

5 Analysis

To better understand the model performance I inspect two contexts containing rare words exemplifying problems of the models.

1. Context with rare word 'regicide':
   Emperor Gegeen Khan, Ayurbarwada’s son and successor, ruled for only two years, from 1321 to 1323. [...] and, after an unsuccessful attempt to calm the princes, he also succumbed to regicide.
   Question: When was Geegen the emperor?
   Answers: BiDAF baseline: 1321 (incorrect). BiDAF + FCM: 1321 to 1323 (correct).

2. Context with rare word 'constancy':
   [...] The rotational inertia of planet Earth is what fixes the constancy of the length of a day and the length of a year. [...] This is why, for example, astronauts experience weightlessness when in free-fall orbit around the Earth, and why Newton’s Laws of Motion are more easily
discernible in such environments. [...]  

Question (a): What do astronaughts experience while in free-fall?  
Answers: BiDAF baseline: no answer (incorrect). BiDAF + FCM: weightlessness (correct).

Question (b): What do astronauts experience when in free-fall orbit around Saturn?  
Answers: BiDAF baseline: no answer (correct). BiDAF + FCM: weightlessness (incorrect).

The first rare-word context - question pair involves numbers in the answer. While the BiDAF baseline without character level embeddings has problems finding the full answer, the character-aware FCM-gate augmented BiDAF captures these type of situations better. Representing OOV words or words with a very low frequency, the FCM-style model will resort to surface embeddings before attempting an answer.

The second rare-word context-question pair highlights a difference of the models in AvNA (answer vs no answer prediction). I observe that the baseline BiDAF too often predicts ‘no answer’ in rare word contexts. In contrast, the FCM - style BiDAFs attempt to answer more often, however, sometimes wrongly as in the case given.

6 Conclusion

I build on the observation of [2] that even strong pre-trained contextual embeddings like BERT have limitations when it comes to rare words. Form-context models (FCM) combining word surface and contextual information have been proposed to improve the rare word performance of contextual word embeddings. I take up this idea and implement an FCM-inspired gating mechanism into the BiDAF for SQuAD v2.0. My analysis focuses on context-question pairs with low average word frequencies or rare words. I use external features like the word frequency obtained from a large corpus as well as part of speech tags to improve the training of the FCM gating mechanism. My results indicate that the FCM-style gating does have superior performance in rare word contexts, however, at the expense of its full sample performance. A combination of these kind of models with overall stronger models could be worthwhile.

Interestingly, the use of external features in the gating itself seems to facilitate the training of the gate parameters. This seems to be especially rewarding for smaller data like SQuAD v2.0. On larger data sets, more expressive features can be expected to be properly learned via end-to-end learning.

References


