# Supplementary Material:
## Natural Language Processing with Small Feed-Forward Networks

**Jan A. Botha** **Emily Pitler** **Ji Ma** **Anton Bakalov**
**Alex Salcianu** **David Weiss** **Ryan McDonald** **Slav Petrov**
Google Inc.
Mountain View, CA
{jabot,epitler,maji,abakalov,salcianu,djweiss,ryanmcd,slav}@google.com

## A Quantization Details

The values comprising a generic embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times D}$ are ordinarily stored with 32-bit floating-point precision in our implementation. For quantization, we first calculate a scale factor $s_i$ for each embedding vector $\mathbf{e}_i$ as

$$s_i = \frac{1}{b-1} \max_j |e_{ij}| .$$

Each weight $e_{ij}$ is then quantized into an 8-bit integer as

$$q_{ij} = \lfloor \frac{1}{2} + \frac{e_{ij}}{s_i} + b \rfloor,$$

where the bias $b = 128$. Hence, the number of bits required to store the embedding matrix is reduced by a factor of 4, in exchange for storing the $V$ additional scale values. At inference time, the embeddings are dequantized on-the-fly.

## B FLOPs Calculation

The product of $\mathbf{A} \in \mathbb{R}^{P \times Q}$ and $\mathbf{b} \in \mathbb{R}^{Q}$ involves $P(2Q-1)$ FLOPs, and our single ReLu hidden layer requires performing this operation once per timestep ($P$=$M$, $Q$=$H_0$). Here, $H_0$ denotes the size of the embedding vector $\mathbf{h}_0$, which equals 408, 464 and 260 for our respective POS models as ordered in Table 2.

In contrast, each LSTM layer requires eight products per timestep, and the BTS model has four layers ($P$=$Q$=320). The particular sequence-to-sequence representation scheme of Gillick et al. (2016) requires at least four timesteps to produce a meaningful output: the individual input byte(s), and a start, length and label of the predicted span. A single timestep is therefore a relaxed lower bound on the number of FLOPs needed for BTS inference.

## C Word Clusters

The word clusters we use are for the 250k most frequent words from a large unannotated corpus that was clustered into 256 classes using the distributed Exchange algorithm (Uszkoreit and Brants, 2008) and the procedure described in Appendix A of Täckström et al. (2012).

The space required to store them in a Bloom map is calculated using the formula derived by Talbot and Talbot (2008): each entry requires $1.23 * (\log \frac{1}{\epsilon} + H)$ bits, where $H$ is the entropy of the distribution on the set of values, and $\epsilon = 2^{-E}$, with $E$ the number of error bits employed. We use 0 error bits and assume a uniform distribution for the 256 values, i.e. $H = 8$, hence we need 9.84 bits per entry, or 300KB for the 250k entries.

## D Lang-ID Details

In our language identification evaluation, the 1,2,3,4-gram embedding vectors each have 6 or 16 dimensions, depending on the experimental setting. Their hashed vocabulary sizes ($V_g$) are 100, 1000, 5000, and 5000, respectively. The hidden layer size is fixed at $M$=208.

We preprocess data by removing non-alphabetic characters and pieces of markup text (i.e., anything located between $<$ and $>$, including the brackets). At test time, if this results in an empty string, we skip the markup removal, and if that still results in an empty string, we process the original string. This procedure is an artefact of the Wikipedia dataset, where some documents contain only punctuation or trivial HTML code, yet we must make predictions for them to render the results directly comparable to the literature.

## E POS Details

The Small FF model in the comparison to BTS uses 2,3,4-grams and some byte unigrams (see fea-

| | |
|---|---|
| bytes | $\forall i \in [0,1], \forall j \in [0,3] : l_{\pm i}^{\pm j}$ |
| char *n*-grams | $\forall i \in [0,3], \forall N \in [2,4] : \{u_{\pm i}^{(N)}\}$ |
| clusters | $\forall i \in [0,3] : c_{\pm i}$ |

Table i: **POS tagging feature templates.** $i$ is a position relative to the focus token. $l_j$ is the value of the $j$-th UTF8 byte from the start/end of a word. $\{u^{(N)}\}$ designates the set of Unicode character *n*-grams in a word. $c$ is the cluster id of a word.

| | |
|---|---|
| char | $\forall i \in [0,1] : \sigma_{\pm i}$.c; $\forall i \in [0,2]\ \beta_{\pm i}$.c |
| bigram | $\forall i \in [0,1] : \sigma_{\pm i}$.b; $\beta_{\pm i}$.b |

Table ii: **Word segmentation feature templates.** '$\beta_{\pm i}$' denotes starting at the $i$-th character to the left/right of the front of the buffer. 'c' and 'b' denote character and character-bigram, respectively.

ture templates in Table i). The *n*-grams have embedding sizes of 16 and the byte unigrams get 4 dimensions. In our $\frac{1}{2}$-dimension setting, the aforementioned dimensions are halved to 8 and 2.

Cluster features get embedding vectors of size 8. The hashed feature vocabularies for *n*-grams are 500, 200, and 4000, respectively. The hidden layer size is fixed at $M$=320.

## F Segmentation Details

Feature templates used in segmentation experiments are listed in Table ii. Besides, we define length feature to be the number of characters between top of $\sigma$ and the front of $\beta$, this maximum feature value is clipped to 100. The length feature is used in all segmentation models, and the embedding dimension is set to 6. We set the cutoff for both character and character-bigrams to 2 in order to learn unknown character/bigram embeddings. The hidden layer size is fixed at $M$=256.

## G Preordering Details

The feature templates for the preorderer look at the top four spans on the stack and the first four spans in the buffer; for each span, the feature templates look at up to the first two words and last two words within the span. The "vanilla" variant of the preorderer includes character *n*-grams, word bytes, and whether the span has ever participated in a SWAP transition. The POS features are the predicted tags for the words in these positions. Table iii shows the full feature templates for the preorderer.

| Features | Positions |
|---|---|
| char bigrams | for $i \in [0,1]\ \sigma(i)_1$ |
| | for $i \in [0,2]\ \sigma(i)_{l_{\sigma(i)}}$ |
| | $\beta(0)_1$ |
| bytes | for $i \in [0,1]\ \sigma(i)_1$ |
| | for $i \in [0,2]\ \sigma(i)_{l_{\sigma(i)}}$ |
| | $\beta(0)_1$ |
| has-swapped | for $i \in [0,1] \sigma(i)$ |
| tags-main | for $i \in [0,1]\ \sigma(i)_1$ |
| | for $i \in [0,2]\ \sigma(i)_{l_{\sigma(i)}}$ |
| | $\beta(0)_1$ |
| tags-aux | for $i \in [0,1]\ \sigma(i)_2\ \sigma(i)_{l_{\sigma(i)-1}}$ |
| | for $i \in [2,3]\ \sigma(i)_1\ \sigma(3)_{l_{\sigma(3)}}$ |
| | $\beta(0)_2\ \beta(0)_{l_\beta(0)-1}\ \beta_{l_{\beta(0)}}$ |
| | for $j \in [1,3]\ \beta(j)_1\ \beta(j)_{l_\beta(j)}$ |

Table iii: **Preordering feature templates.** Each feature group applies to the set of positions given. $\sigma(i)$ denotes the $i$-th span from the top of the stack, and $\beta(j)$ the $j$-th span from the front of the buffer. Within a span $s$, the $l_s$ tokens are $s_1 ... s_{l_s}$, so $s_1$ is the leftmost token in $s$ and $s_{l_s}$ the rightmost.

| *Model.* | *L.R.* | *Mom.* | $\gamma$ | *Steps* | *D.P.* |
|---|---|---|---|---|---|
| C-64 | 0.03 | 0.8 | 32K | 3.8M | 0.2 |
| C-256 | 0.03 | 0.8 | 32K | 3.6M | 0.4 |
| C-64+B-04 | 0.03 | 0.8 | 64K | 7.6M | 0.3 |

Table iv: **Segmentation:** Optimal hyperparameter settings per model for our segmentation experiments reported in Table 4. The columns show learning rate (L.R.), momentum factor (Mom.), the step-frequency at which the learning rate is scaled by 0.96 ($\gamma$), and the number of steps at which training was stopped because accuracy peaked on the held-out tuning data. The column *D.P.* shows the optimal dropout probability.

| | *L.R.* | *Mom.* | $\gamma$ | *Steps* |
|---|---|---|---|---|
| No POS tags | 0.05 | 0.9 | 2k | 38k |
| w/ POS tags | 0.05 | 0.9 | 8k | 46k |
| *POS model* | 0.05 | 0.9 | 8k | 500k |
| w/ tagger input fts. | 0.1 | 0.8 | 4k | 76k |

Table v: **Preordering:** Optimal hyperparameter settings obtained for our preordering experiments reported in Table 6. Columns have the same meanings as in Table iv.

| # | Small FF 6 dim | | | Small FF 16 dim | | |
|---|---|---|---|---|---|---|
| | *L.R.* | *Mom.* | $\gamma$ | *L.R.* | *Mom.* | $\gamma$ |
| 0 | 0.4 | 0.9 | 8k | 0.4 | 0.9 | 16k |
| 1 | 0.4 | 0.9 | 32k | 0.4 | 0.9 | 32k |
| 2 | 0.4 | 0.9 | 32k | 0.4 | 0.9 | 8k |
| 3 | 0.3 | 0.9 | 64k | 0.5 | 0.9 | 16k |
| 4 | 0.4 | 0.8 | 100k | 0.4 | 0.9 | 32k |
| 5 | 0.5 | 0.8 | 100k | 0.4 | 0.9 | 64k |
| 6 | 0.3 | 0.9 | 32k | 0.3 | 0.9 | 32k |
| 7 | 0.3 | 0.9 | 100k | 0.5 | 0.9 | 16k |
| 8 | 0.4 | 0.9 | 32k | 0.5 | 0.9 | 8k |
| 9 | 0.4 | 0.9 | 32k | 0.3 | 0.9 | 16k |

Table vi: **Lang-ID:** Optimal hyperparameter settings obtained with the results reported in Table 1. The first column is the cross-validation fold, while the other columns have the same meanings as in Table iv.

## References


Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of NAACL-HLT*, pages 1296–1306, San Diego, USA. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487.

David Talbot and John Talbot. 2008. Bloom maps. In *Proceedings of the Meeting on Analytic Algorithmics and Combinatorics*, pages 203–212. Society for Industrial and Applied Mathematics.

Jakob Uszkoreit and Thorsten Brants. 2008. Distributed Word Clustering for Large Scale Class-Based Language Modeling in Machine Translation. In *ACL*, pages 755–762.


| *Lang.* | *L.R.* | *Mom.* | $\gamma$ | *Steps* | *Acc.* |
|---|---|---|---|---|---|
| **Small FF** | | | | | |
| bg | 0.1 | 0.8 | 32k | 90k | 97.12 |
| cs | 0.05 | 0.9 | 32k | 480k | 97.97 |
| da | 0.05 | 0.9 | 32k | 480k | 94.17 |
| en | 0.01 | 0.9 | 128k | 660k | 92.50 |
| fi | 0.05 | 0.9 | 8k | 210k | 93.84 |
| fr | 0.1 | 0.8 | 64k | 60k | 95.10 |
| de | 0.1 | 0.8 | 8k | 120k | 91.23 |
| el | 0.08 | 0.9 | 64k | 60k | 96.88 |
| id | 0.08 | 0.8 | 32k | 180k | 91.60 |
| it | 0.08 | 0.8 | 128k | 330k | 96.79 |
| fa | 0.08 | 0.9 | 128k | 60k | 95.80 |
| es | 0.1 | 0.8 | 32k | 60k | 94.37 |
| sv | 0.1 | 0.9 | 8k | 210k | 94.54 |
| **Small FF + Clusters** | | | | | |
| bg | 0.08 | 0.8 | 64k | 120k | 97.72 |
| cs | 0.1 | 0.8 | 16k | 420k | 98.12 |
| da | 0.1 | 0.8 | 32k | 360k | 95.49 |
| en | 0.05 | 0.8 | 8k | 510k | 93.88 |
| fi | 0.1 | 0.8 | 8k | 300k | 94.97 |
| fr | 0.05 | 0.9 | 8k | 630k | 95.65 |
| de | 0.05 | 0.9 | 8k | 480k | 92.40 |
| el | 0.1 | 0.9 | 8k | 60k | 97.60 |
| id | 0.1 | 0.8 | 64k | 150k | 91.94 |
| it | 0.1 | 0.8 | 32k | 270k | 97.36 |
| fa | 0.08 | 0.9 | 64k | 90k | 96.24 |
| es | 0.05 | 0.9 | 128k | 30k | 95.01 |
| sv | 0.08 | 0.9 | 16k | 150k | 95.90 |
| **Small FF ($\frac{1}{2}$ Dim.) + Clusters** | | | | | |
| bg | 0.1 | 0.8 | 128k | 210k | 97.76 |
| cs | 0.05 | 0.9 | 32k | 420k | 98.06 |
| da | 0.05 | 0.9 | 16k | 240k | 95.33 |
| en | 0.05 | 0.8 | 8k | 300k | 93.06 |
| fi | 0.05 | 0.9 | 16k | 390k | 94.66 |
| fr | 0.08 | 0.9 | 128k | 120k | 95.28 |
| de | 0.08 | 0.9 | 16k | 90k | 92.13 |
| el | 0.08 | 0.9 | 16k | 60k | 97.42 |
| id | 0.08 | 0.9 | 8k | 690k | 92.15 |
| it | 0.05 | 0.9 | 64k | 210k | 97.42 |
| fa | 0.1 | 0.8 | 8k | 510k | 96.19 |
| es | 0.08 | 0.9 | 8k | 60k | 94.79 |
| sv | 0.1 | 0.8 | 16k | 300k | 95.76 |

Table vii: **POS:** Optimal hyperparameter settings per language obtained for our POS experiments. Columns have the same meanings as in Table iv. The final column shows the test set accuracies that back the averages shown in Table 2.