1 Summary

This is a note for [1]. The implementation code for this paper can be found at [1].

The important components are as follows:

1. RNN language model
2. Morphological priors
3. Latent word embedding b_w.
4. Morpheme emebedding u_m.
5. Variational distribution $Q(b)$

2 Latent Word Embedding and Morpheme Embedding

Each morpheme is segmented in unsupervised fashion according to Morfessor. For example, $u_{-ism} = (-0.24, 5, -111)$.

When inferring $P(x)$, we will have to infer $P(b)$ too since $P(b)$ appears in the lower variational bound.

$$b_{w,i} \sim Bernouli(sigmoid(\sum_{m \in M_w} u_{m,i}))$$

i.e. for outcomes or the range of a probabilistic variable $b_{w,i}$ is either 0 or 1,

$$P(b_{w,i}) = sigmoid(\sum_{m \in M_w} u_{m,i})^{b_{w,i}}(1 - sigmoid(\sum_{m \in M_w} u_{m,i}))^{1-b_{w,i}}$$

So let's look into an example. Let $M = perfection, -ism$ $u_{perfection} = (0, -1.1, 1)$
$u_{-ism} = (2, 5.1, 3)$
When $w = perfectionism$, then
$b_{w,0} \sim Bernoulli(sigmoid(0 + 2)) \approx 0.88$
$b_{w,1} \sim Bernoulli(sigmoid(-1.1 + 5.1)) \approx 0.98$
$b_{w,2} \sim Bernoulli(sigmoid(1 + 3)) \approx 0.98$
So $P(b_w = (1, 1, 1)) = 0.88 * 0.98 * 0.98 \approx 0.84$.

3 Hidden state

The hidden state at time h_t (vector) is

$$h_t = sigmoid(\Theta h_{t-1} + b_{x_t})$$

where x_t is the word corresponding to the position t, and Θ is the parameter for the recurrence function (recurrent weights[2]).

[1] https://github.com/rguthrie3/MorphologicalPriorsForWordEmbeddings
[2] http://peterroelants.github.io/posts/rnn_implementation_part01/

4 What is going on inside $D_{KL}(Q(b)||P(b))$?

$$D_{KL}(q(b_{w,i})||P(b_{w,i})) = q(b_{w,i})\log(\frac{q(b_{w,i})}{P(b_{w,i})})$$

$$= q(b_{w,i})(\log(q(b_{w,i})) - \log(P(b_{w,i}))$$

$$= E_q[\log(q(b_{w,i}))] - E_q[\log(P(b_{w,i}))]$$

$$E_q[\log(q(b_{w,i};\gamma_{w,i}))] = q(b_{w,i} = 1) * \log(\gamma_{w,i}) + q(b_{w,i} = 0) * \log(1 - \gamma_{w,i})$$

$$= \gamma_{w,i} * \log(\gamma_{w,i}) + (1 - \gamma_{w,i}) * \log(1 - \gamma_{w,i})$$

$$E_q[\log P(b_{w,i})] = q(b_{w,i} = 1) * \log(sigmoid(\sum_{m \in M_w} u_{m,i})) + q(b_{w,i} = 0) * \log((1 - sigmoid(\sum_{m \in M_w} u_{m,i})))$$

$$= \gamma_{w,i} * \log(sigmoid(\sum_{m \in M_w} u_{m,i})) + (1 - \gamma_{w,i}) * \log((1 - sigmoid(\sum_{m \in M_w} u_{m,i})))$$

Note that 'morpho_level_reps = (self.morpho_embed_lookup.apply(morpho_idxs) * masks).sum(axis=2)' reprepresnts $\sum_{m \in M_w} u_{m,i}$

$$1 - sigmoid(x) = \frac{1 + e^{-x}}{1 + e^{-x}} - \frac{1}{1 + e^{-x}} = \frac{e^{-x}}{1 + e^{-x}}$$
$$= \frac{1}{e^x + 1}$$
$$\log(1 - sigmoid(x)) = \log(\frac{1}{e^x + 1}) = -\log(e^x + 1)$$

References

[1] Parminder Bhatia, Robert Guthrie, and Jacob Eisenstein. Morphological priors for probabilistic neural word embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 490–500, Austin, Texas, November 2016. Association for Computational Linguistics.