A. Derivation of the gradient with respect to the knowledge retriever

We compute the gradient of the REALM pre-training objective (a log-likelihood) with respect to the parameters of the knowledge retriever, θ:

$$\begin{aligned}\nabla \log p(y \,|\, x) &= p(y \,|\, x)^{-1} \nabla p(y \,|\, x) \\ &= p(y \,|\, x)^{-1} \sum_z p(y \,|\, z, x) \nabla p(z \,|\, x) \\ &= p(y \,|\, x)^{-1} \sum_z p(y \,|\, z, x) p(z \,|\, x) \nabla \log p(z \,|\, x) \\ &= \sum_z p(z \,|\, y, x) \nabla \log p(z \,|\, x),\end{aligned}$$

where the last line follows from applying conditional Bayes' rule. We can then expand $\nabla \log p\,(z \,|\, x)$ as:

$$\begin{aligned}\nabla \log p(z \,|\, x) &= \nabla \log \frac{\exp f(x, z)}{\sum_{z'} \exp f(x, z')} \\ &= \nabla \left[f(x, z) - \log \sum_{z'} \exp f(x, z') \right] \\ &= \nabla f(x, z) - \sum_{z'} p(z' \,|\, x) \nabla f(x, z')\end{aligned}$$

Plugging this back into the first set of equations yields:

$$\begin{aligned}\nabla \log p\,(y \,|\, x) &= \sum_z p\,(z \,|\, y, x) \left[\nabla f(x, z) - \sum_{z'} p\,(z' \,|\, x)\, \nabla f(x, z') \right] \\ &= \sum_z p\,(z \,|\, y, x)\, \nabla f(x, z) - \sum_{z'} p\,(z' \,|\, x)\, \nabla f(x, z') \\ &= \sum_z \left[p\,(z \,|\, y, x) - p\,(z \,|\, x) \right] \nabla f(x, z) \\ &= \sum_z \left[\frac{p\,(y \,|\, z, x)\, p\,(z \,|\, x)}{p\,(y \,|\, x)} - p\,(z \,|\, x) \right] \nabla f(x, z) \\ &= \sum_z \left[\frac{p\,(y \,|\, z, x)}{p\,(y \,|\, x)} - 1 \right] p\,(z \,|\, x)\, \nabla f(x, z).\end{aligned}$$

In the second line, we used the fact that the overall expression is an expectation with respect to $p\,(z \,|\, y, x)$, and the terms which depend on z' but not z can be moved out of that expectation.

B. Connection between REALM and supervised learning

From the equations in Appendix A, we saw that

$$\nabla \log p\,(y \,|\, x) = \sum_z \left[p\,(z \,|\, y, x) - p\,(z \,|\, x) \right] \nabla f(x, z).$$

Suppose that there exists one document z^* which causes the model to achieve perfect prediction accuracy (i.e., $p\,(y \,|\, z^*, x) = 1$), while all other documents z' result in zero accuracy (i.e., $p\,(y \,|\, z', x) = 0$). Under this setting, $p\,(z^* \,|\, y, x) = 1$ (provided that $p\,(z^* \,|\, x)$ is non-zero), which causes the gradient to become

$$\begin{aligned}\nabla \log p\,(y \,|\, x) &= \nabla f\,(x, z^*) - \sum_z p\,(z \,|\, x)\, \nabla f(x, z) \\ &= \nabla \log p\,(z^* \,|\, x)\,.\end{aligned}$$

From this, we see that gradient descent on the REALM objective is equivalent to gradient descent on $\log p\,(z^* \,|\, x)$. This is none other than the typical maximum likelihood training objective used in supervised learning, where z^* is the "gold" document.

C. Adapting to new knowledge

An explicit retrieval system allows us to adapt to new world knowledge simply by modifying the corpus documents. To demonstrate this ability, we replace the knowledge corpus with a more recent version of Wikipedia corpus after pre-training is done. When the input query is about a fact where the two corpora disagree, REALM can change the prediction to reflect the updated information, as exemplified in Table 4. However, even with an explicit retrieval mechanism, the knowledge-augmented encoder will still end up remembering some world knowledge, making the prediction of some input sentences not updated with the new corpus. (For instance, the model predicts "`Thatcher`" for "`___ is the prime minister of United Kingdom.`" on both corpora, perhaps due to the frequent mention of her name in Wikipedia articles.)

D. Retrieval Utility

The null document $\varnothing$ described in Section 3.4 provides a way to measure the importance of a retrieved document z: we define the *retrieval utility* (RU) of z for the masked input x as the difference between the log-likelihood of the knowledge-augmented encoder when conditioning on z versus on $\varnothing$:

$$\mathrm{RU}(z \,|\, x) = \log p(y \,|\, z, x) - \log p(y \,|\, \varnothing, x). \qquad (2)$$

A negative RU shows that z is less useful for predicting y than the null document. This could mean that z is irrelevant to x, but could also mean that the masked tokens in x do not require world knowledge to predict, or that the world knowledge is sufficiently commonplace it has been baked into the model's parameters. In practice, we find that RU increases steadily over the course of pre-training, and is more predictive of good performance on the downstream task of Open-QA than even the overall log-likelihood. An example of how RU behaves over time and across different settings is in Figure 4.

x:	“Jennifer ___ formed the production company Excellent Cadaver.”
BERT	also (0.13), then (0.08), later (0.05), . . .
REALM ($\mathcal{Z}$ =20 Dec 2018 corpus)	smith (0.01), brown (0.01), jones (0.01)
REALM ($\mathcal{Z}$ =20 Jan 2020 corpus)	**lawrence** (0.13), brown (0.01), smith (0.01), . . .

Table 4. An example where REALM adapts to the updated knowledge corpus. The Wikipedia page “Excellent Cadaver” was added in 2019, so the model was not able to recover the word when the knowledge corpus is outdated (2018). Interestingly, the same REALM model pre-trained on the 2018 corpus is able to retrieve the document in the updated corpus (2020) and generate the correct token, “Lawrence”.

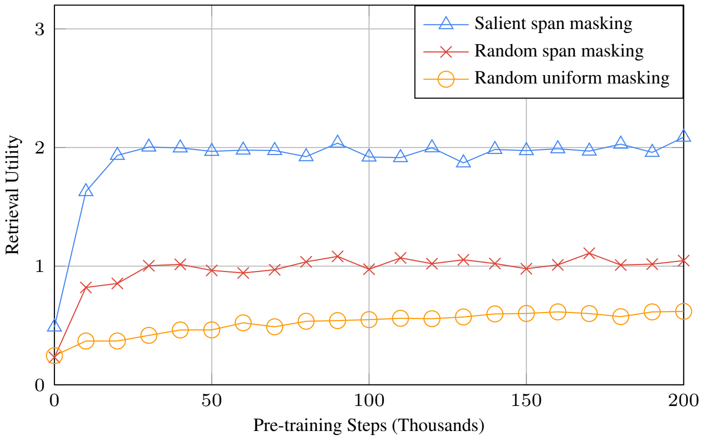

Figure 4. The Retrieval Utility (RU, described in Eq. 2) vs the number of pre-training steps. RU roughly estimates the “usefulness” of retrieval. RU is impacted by the choice of masking and the number of pre-training steps.