Beyond [CLS] through Ranking by Generation

Cicero Nogueira dos Santos, Xiaofei Ma,
Ramesh Nallapati, Zhiheng Huang and Bing Xiang
AWS AI
New York City, NY
{cicnog,xiaofeim,rnallapa,zhiheng,bxiang}@amazon.com

Abstract

Generative models for Information Retrieval, where ranking of documents is viewed as the task of generating a query from a document's language model, were very successful in various IR tasks in the past. However, with the advent of modern deep neural networks, attention has shifted to discriminative ranking functions that model the semantic similarity of documents and queries instead. Recently, deep generative models such as GPT2 and BART have been shown to be excellent text generators, but their effectiveness as rankers have not been demonstrated yet. In this work, we revisit the generative framework for information retrieval and show that our generative approaches are as effective as state-of-the-art semantic similarity-based discriminative models for the answer selection task. Additionally, we demonstrate the effectiveness of unlikelihood losses for IR.

1 Introduction

Most recent approaches for ranking tasks in Information Retrieval (IR) such as passage ranking and retrieval of semantically related questions have focused primarily on discriminative methods using neural networks that learn a similarity function to compare questions and candidate answers (Severyn and Moschitti, 2015; dos Santos et al., 2015; Tan et al., 2016; Tay et al., 2017, 2018). On the other hand, classical literature on probabilistic models for IR showed that language modeling, a type of simple generative model, can be effective for document ranking (Zhai, 2008; Lafferty and Zhai, 2001; Ponte and Croft, 1998). The key idea consists of first training a unique language model lm_i for each candidate document d_i, then using the likelihood of generating the input query using lm_i, denoted by $P(q|lm_i)$, as the ranking score for document d_i.

Recent advances in neural language models (NLMs) have led to impressive improvements in the quality of automatically generated text (Radford et al., 2019). However, to the best of our knowledge, there is no existing work on exploring the effectiveness of modern generative models such as GPT2, for complex ranking tasks such as answer selection. In this work, we intend to fill this gap by demonstrating that large pretrained generative models can be very effective rankers. Unlike classic LM based approaches for IR that employ separate LMs for each document, our proposed method uses a single global LM that applies to all documents. The global pretrained generator is fine-tuned on the task of query generation conditioned on document content as the context. Additionally, in order to leverage both positive and negative examples, we propose the use of (1) unlikelihood loss on negative examples and (2) ranking loss on the likelihood of positive and negative examples. At inference time, given an input query, our method scores each candidate document using the likelihood of generating the query given the document, as estimated by our fine-tuned global LM.

We focus our experiments on the task of answer selection (a.k.a passage ranking). In this task, given an input question and a set of candidate passages, the goal is to rank the candidate passages so that passages containing the correct answer appear at the top of the ranked list. Considerable body of work exists on the use of NNs for this task (Feng et al., 2015; Severyn and Moschitti, 2015; Tan et al., 2016; dos Santos et al., 2016; Rao et al., 2016; Wang et al., 2017), where the most recent ones use BERT-based models that perform discrimination based on the special $[CLS]$ token (Nogueira and Cho, 2019; Li et al., 2019; Xu et al., 2019). A contemporaneous work by Nogueira et al. (2020) also proposes a generative approach for the passage ranking task. However, while their approach decides the relevance of a passage by generating a single keyword (e.g. *true* or *false*), our method

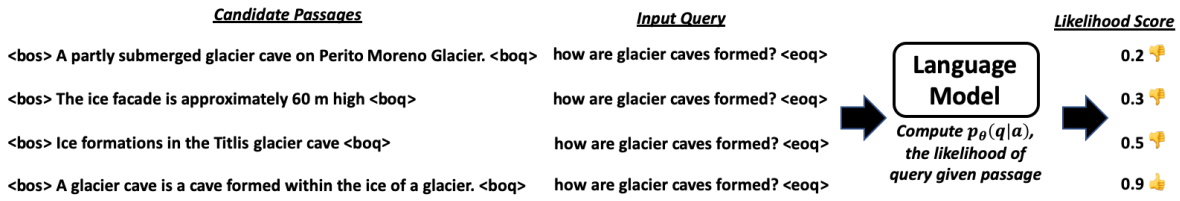

Figure 1: Illustration of the inference step of our *ranking by generation* approach. Each candidate passage a_k is ranked based on the likelihood of generating the question q conditioned on the passage, $p_\theta(q|a_k)$.

uses the conditional likelihood of generating the question given the passage as a relevance score.

We perform extensive experiments using GPT2 (Radford et al., 2019) and BART (Lewis et al., 2019), which are Transformer-based LMs (Vaswani et al., 2017) that were pretrained using large volumes of textual data. The LMs are fine-tuned on four different passage ranking datasets separately: WikipassageQA, WikiQA, InsuranceQA_V2, and YahooQA. Our experimental results indicate that our generative approaches are as effective as state-of-the-art discriminative-based approaches for answer selection.

2 Ranking by Generation

2.1 Background

The goal in language modeling is to learn the probability distribution $p(x)$ over variable-length token sequences $x = (x_1, x_2, ..., x_{|x|})$, where the tokens come from a fixed size vocabulary, $x_i \in V$. When training an LM with *causal language modeling objective*, which consists of predicting the next token by looking at the past only, we can denote this distribution by the conditional probability of the next token given the previous ones (Bengio et al., 2003):

$$p(x) = \prod_{i=1}^{|x|} p(x_i|x_{<i}) \quad (1)$$

GPT2 (Radford et al., 2019) is an example of a state-of-the-art neural LM trained with causal language modeling objective. The usual approach to train an LM using a neural network with parameters θ consists on performing maximum likelihood estimation (MLE) by minimizing the negative log-likelihood over a large text corpus $D = \{x^1, x^2, ..., x^{|D|}\}$, where each x_k is a document of length $|x_k|$:

$$\mathcal{L}(D) = -\sum_{k=1}^{|D|}\sum_{i=1}^{|x^k|} \log p_\theta(x_i^k|x_{<i}^k) \quad (2)$$

Conditional LMs are a simple extension of regular LMs where the generation is conditioned on some additional context c (Keskar et al., 2019):

$$p(x|c) = \prod_{i=1}^{|x|} p(x_i|x_{<i}, c) \quad (3)$$

2.2 Proposed Ranking Approach

Our proposed approach for passage ranking by generation consists of first fine-tuning a pretrained large LM on the task of question generation conditioned on the passage, using the conditional LM approach shown in Eq. 3. In practice, each input for the fine-tuning step is as follows:

`<bos> passage <boq> question <eoq>`

where the *passage* is considered as a prompt, and the log-likelihood used in the training comes only from the tokens starting after the keyword *<boq>*, since we use the passage as a conditioning context. In other words, at training time, we minimize the negative conditional log-likelihood $-\log P(q|a)$, where a is a passage relevant to the query q. At inference time, given a query q, our conditional LM scores each candidate passage a_k using the likelihood of generating the question conditioned on the passage, $s(a_k) = p_\theta(q|a_k)$. Fig. 1 illustrates the inference step of our proposed approach.

2.3 Unlikelihood Loss for Ranking

Datasets for training passage rankers normally contain both positive and negative examples. Therefore, it is natural to use both types of examples in order to leverage all the available data. Let D be the set of examples (q, a, y), where y is 1 if the passage a is a positive answer for q, or 0 otherwise. We fine-tune the LM using the following loss function:

$$\mathcal{L}(D) = -\sum_{(q,a,y)\in D}\sum_{i=1}^{|q|} y\log(p_\theta(q_i|q_{<i}, a)) + (1-y)\log(1-p_\theta(q_i|q_{<i}, a)) \quad (4)$$

The second term in Eq. 4 resembles the unlikelihood training objective of Welleck et al. (2019). However, while we use an unlikelihood objective with the aim of teaching the LM which questions are unlikely given the passage, Welleck et al. (2019) use an unlikelihood objective with the aim of improving text generation. We use the acronym LUL to refer to the loss function in Eq. 4, which performs likelihood and unlikelihood estimation.

We experimented an additional loss function to fine-tuning the LMs which consists on imposing a pairwise ranking loss on the likelihood (RLL) of positive and negative examples as follows:

$$\begin{aligned}\mathcal{L}(D) &= \sum_{(q,a+,a-)\in D} \max\{0, \lambda - \log p_\theta(q|a^+) \\ &+ \log p_\theta(q|a^-)\} \end{aligned} \quad (5)$$

The use of unlikelihood losses to penalize negative examples is a natural choice for fine-tuning generative models. Note that Eq. 4 is an extension of the regular cross-entropy loss where we just added the unlikelihood term, while Eq. 5 is its ranking-based (hinge loss) version. The unlikelihood term in Eq. 4 can also be seen as a regularizer, which makes the ranking model less overconfident when computing query likelihoods.

3 Experiments and Discussion

3.1 Datasets

We use four different publicly available answer selection datasets in our experiments: WikipassageQA (Cohen et al.), WikiQA (Yang et al., 2015), InsuranceQA_V2 (Feng et al., 2015), and YahooQA (Tay et al., 2017). Statistics about the datasets are shown in Table 1. The four datasets also provide validation sets, which have size similar to the respective test sets.

Dataset	Train: #Q (#P/Q)	Test: #Q (#P/Q)
WikiQA	873 (9)	243 (9)
WikipassageQA	3,332 (58.3)	416 (57.6)
InsuranceQA	12,889 (500)	2,000 (500)
YahooQA	50,112 (5)	6,283 (5)

Table 1: Dataset statistics. #Q stands for *number of questions* and #P/Q is the *average number of passages per question*

3.2 Language Model Setup

We use pretrained GPT2-base (12 layers, 117M parameters), GPT2-large (24 layers, 345M params), BART-base (6 layers encoder and 6 layers decoder, 139M params) and BART-large (12 layers encoder and 12 layers decoder, 406M params) models in our experiments. We adopted the implementation and pretrained models from Wolf et al. (2019). We fine-tune GPT2 and BART on each training dataset separately. We perform a maximum of 10 fine-tuning epochs and adopt early stopping using the validation sets. Most of the hyperparmeters used for fine-tuning are the default ones from Wolf et al. (2019)[1], except for learning rate for BART, which we set to $1e-5$.

In the experiments presented below, the subscript MLE corresponds to models fine-tuned using just maximum likelihood estimation (Eq. 2), which means that only positive examples are used. The subscript LUL corresponds to models fine-tuned using maximum likelihood and unlikelihood estimation (Eq. 4), while RLL are models fine-tuned using the ranking loss in Eq. 5. For MLE and LUL, we use a mini-batch size of 64 for InsuranceQA and 32 for the other 3 datasets. The number of negative examples per positive examples is set to 5 in the case of LUL.

When fine-tuning with RLL loss (Eq. 5), we use a batch size of 8. During training, when processing a question we randomly sample 15 negative passages from the set of negative passages of the question. However, only the negative passage with the highest score is used to update the model. Early experiments demonstrated that this strategy performs similarly to the usual pairwise approach.

3.3 Ranking Results

In Table 2 we present the experimental results for our proposed generative approach and four state-of-the-art discriminative baselines, which are based on BERT (Devlin et al., 2019) and BART. Both BERT-Sel (Li et al., 2019) and BERT-PR (Xu et al., 2019) fine-tuned BERT-base using a ranking loss on the score computed with $[CLS]$ token. We trained a BERT-large model using $[CLS]$-based scoring + ranking loss (rows 3). We additionally trained a discriminative version of BART-large (row 4) where the input for the encoder and the decoder are the passage and the question, respectively. As it is normally adopted in BART for classification (Lewis et al., 2019), we take the representation generated by the decoder for the last token and use it to create a score by applying a linear layer. Such as the discriminative BERT models, we also optimize

[1] https://github.com/huggingface/transformers/blob/master/examples/run_lm_finetuning.py

ID	Dataset	YahooQA			WikiQA			WikipassageQA			InsuranceQA		
		MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1	MAP	MRR	P@1
Discriminative Approaches													
1	BERTSel-base (Li et al., 2019)	.942	.942	-	.753	.77	-	-	-	-	-	-	-
2	BERT-PR-base (Xu et al., 2019)	-	-	-	-	-	-	.735	.809	.702	.413	.496	.401
3	BERT-PR-large	.965	.965	.939	.844	.856	.765	.775	.838	.748	.410	.492	.394
4	BART-large	.967	.967	.943	.845	**.861**	.765	.803	**.866**	**.789**	.435	.518	.423
Generative Approaches													
5	GPT2-base [no fine-tuning]	.499	.499	.265	.516	.522	.337	.215	.250	.132	.050	.071	.034
6	GPT2-base$_{MLE}$	.768	.768	.631	.550	.555	.354	.654	.738	.632	.430	.516	.428
7	GPT2-base$_{LUL}$ (ours)	.905	.905	.905	.690	.701	.547	.723	.807	.716	.427	.512	.422
8	GPT2-base$_{RLL}$ (ours)	.958	.958	.928	.774	.792	.683	.735	.810	.704	.414	.494	.397
9	BART-base$_{LUL}$ (ours)	.928	.928	.876	.778	.788	.658	.738	.813	.719	.440	.526	.434
10	BART-base$_{RLL}$ (ours)	.961	.961	.934	.775	.792	.654	.761	.834	.743	.422	.503	.408
11	GPT2-large$_{LUL}$ (ours)	.917	.917	.857	.736	.742	.609	.755	.825	.738	.444	.532	.439
12	GPT2-large$_{RLL}$ (ours)	.954	.954	.922	.819	.834	.733	.755	.831	.728	.408	.489	.389
13	BART-large$_{LUL}$ (ours)	.949	.949	.911	.802	.815	.712	.789	.848	.764	**.465**	**.553**	**.461**
14	BART-large$_{RLL}$ (ours)	**.970**	**.970**	**.948**	**.849**	**.861**	**.769**	**.808**	**.867**	**.791**	.444	.529	.433

Table 2: Experimental results for different passage ranking models and datasets.

BART-large using a ranking loss. The performance of the passage ranking models is assessed using the metrics Mean Average Precision (MAP), Mean Reciprocal Rank (MRR) and Precision at 1 (P@1). Scores are computed with the official *trec_eval* tool.

In the middle part of Table 2, we compare GPT2-base without any fine-tuning (row 5), and finetuned with either MLE (6), LUL (7) or RLL (8). When the pretrained model only is used (no fine-tuning) the results are very poor. Which is understandable, given that the pattern of having a passage followed by a question might not be very recurrent in the data used to pretrain GPT2. Comparing MLE (row 6) with LUL (row 7), we see that the inclusion of the unlikelihood term (Eq. 4) has a significant positive impact for all datasets but InsuranceQA. We believe the unlikelihood loss does not help on InsuranceQA because this dataset was not human curated and therefore contains a significant number of false-negative examples, which can hurt performance when used to compute the unlikelihood loss. Compared to BERT-base models, GPT2-base$_{LUL}$ is very competitive for most of the datasets except WikiQA, while GPT2-base$_{RLL}$ demonstrates more robust results across the different datasets. In rows 9 and 10 we show results for BART-base, where we see similar trends to GPT2-base with regard to LUL and RLL losses. BART-base$_{RLL}$ is overall better than BART-base$_{LUL}$ and GPT2-base models.

In the bottom part of Table 2, we also show results for GPT2-large and BART-large using LUL and RLL (rows 11 to 14). Overall, the larger generative models do a better job than the smaller ones, as expected. Among the generative approaches, BART-large$_{RLL}$ (row 14) is the model that performs the best for most of the datasets. We believe that BART-based generative models outperform GPT2-based models due to 1) the larger number of pretraining tasks used in BART and 2) the use of bidirectional attention in the encoder side (which processes the passage). Comparing BART-large$_{RLL}$ with discriminative BART-large (row 4), we can see that BART-large$_{RLL}$ produces better results for InsuranceQA, while achieving similar performance for YahooQA, WikiQA and WikipassageQA. Overall, our proposed generative approach produces state-of-the-art results on the four tested datasets in all metrics.

Model	Likelihood	MAP	MRR	P@1
GPT2-base$_{LUL}$	$p_\theta(q\|a)$	.723	.807	**.716**
GPT2-base$_{LUL}$	$p_\theta(a\|q)$	.414	.464	.259
GPT2-base$_{RLL}$	$p_\theta(q\|a)$	**.735**	**.810**	.704
GPT2-base$_{RLL}$	$p_\theta(a\|q)$	.531	.617	.478

Table 3: Experimental results on using passage vs. question as the conditional context. Results are computed on the WikipassageQA dataset

3.4 Ranking with Passage Likelihood

A different setup that can be used for our approach is to compute the likelihood of the passage given the question, where the score for a candidate passage a^k is given by $s(a_k) = p_\theta(a_k|q)$, and the score needs to be normalized by the passage length $|a_k|$. This setup is inherently more difficult because the passage is normally much longer than the question and might contain many tokens that are not relevant for the question.

In Table 3, we present experimental results where we compare the use of either passage or question as the conditional context. As expected, using the likelihood of the passage given the question ($p_\theta(a_k|q)$) as the score results in worse perfor-

Passage	This phenomenon happens usually in the winter. In 2013, Sao Paulo was the most populous city in Brazil and in South America. According to the 2010 IBGE Census, there were 11,244,369 people residing in the city of Sao Paulo. The census found 6,824,668 White people , 3,433,218 Pardo people , 736,083 Black people , 246,244 Asian people and 21,318 Amerindian people . In 2010, the city had 2,146,077 opposite-sex couples and 7,532 same-sex couples. The population of Sao Paulo was 52.6% female and 47.4% male.
Generated Questions	How diverse are the demographics of the city of Sao Paulo? How diverse is the population of the city of Sao Paulo? *Why are people living in the city of Sao Paulo?*
Passage	The presidential electors cast their ballots for President and Vice President, and Congress tallied these votes on January 8, 2009. The voter turnout for this election was broadly predicted to be high by American standards, and a record number of votes were cast. The final tally of total votes counted was 131.3 million, compared to 122.3 million in 2004 . ... . Another estimate puts the eligible voter population at 213,313,508, resulting in a turnout rate of 61.6%, which would be the highest turnout rate since 1968.
Generated Questions	What are the demographics and voting trends of the US presidential elections? What is a voter turnout like in the U.S. presidential elections? *What factors contributed to high voter turnout in 2008?* *Why might voters have difficulty voting?*

Table 4: Examples of automatically generated questions using the GPT2-large$_{LUL}$ model fine-tuned on the WikipassageQA dataset with likelihood $p_\theta(q|a)$. The passages were extracted from the test set.

Question	What was the election turnout for 2008?
Generated Passage	The turnout in the 2008 general election was the highest in the United Kingdom since the 1920s, and the highest turnout in the history of the UK. The turnout was higher than that for the general election of the previous year. The 2008 general election was a close contest, with the Conservatives winning a majority of seats and the Labour Party winning a majority of seats in Parliament. The Conservative Party won the largest majority in Parliament, with 332 of 538 seats, and the Liberal Democrats gained the largest number of seats in the House of Commons, winning 232...
Generated Passage	The turnout was higher in the two states that had been held for the presidency by Barack Obama. The Democratic Party won the state of New Hampshire, the state that Obama won by more than 2 million votes, and the Republican Party won the state of Ohio by nearly 2.5 million votes. In the 2008 presidential election, Obama defeated Mitt Romney by more than 3 million votes, winning 332 of the 435 electoral votes. The election was the second of four presidential elections. The results of the election were announced on November 4. The election was the first in ...

Table 5: Examples of automatically generated passages using the GPT2-large$_{LUL}$ model fine-tuned on the WikipassageQA dataset with likelihood $p_\theta(a|q)$. The question was extracted from the test set.

mance for both fine-tuning approaches: LUL and RLL.

3.5 Question and Passage Generation

A good side effect of using generative models to perform ranking is that we can use the trained model to generate new questions given a passage and vice-versa (depending on the conditioning context used for fine-tuning). This type of synthetically generated data could be used as additional training data to improve discriminative models such as BERT-PR (Xu et al., 2019). In Tables 4 and 5, we present some examples of questions and passages, respectively, that were generated using our fine-tuned GPT2-large$_{LUL}$ LM. In both cases we use a mixture of top k-sampling (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2019) to generate the samples. Please note that the passages in Table 4 were extracted from the test set and are not present in the training set. The same applies for the question used in Table 5.

In Table 4, we can see that the generated questions are very fluent and, for most questions (except for the ones in *italic*), the input passage contains the answer for the question. In Table 5, we can observe that the generated passages are quite related to the input question. However, the content is normally not factual and contains inconsistencies and some repetitions.

4 Conclusion

We have proposed a new generative approach for IR based on large pretrained neural language models, and demonstrated their effectiveness as rankers by providing robust experimental results on four different datasets. Additionally, we demonstrated that unlikelihood-based losses are effective for allowing the use of negative examples in generative-based information retrieval. We believe that our approach can also be effectively used for text classification problems, where the score of a class label c is computed as the likelihood of generating the class label c given the document d, $p(c|d)$.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Daniel Cohen, Liu Yang, and W. Bruce Croft. Wikipas-

sageqa: A benchmark collection for research on non-factoid answer passage retrieval. In *41st International ACM SIGIR Conference*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, Minneapolis, Minnesota.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. *CoRR*, abs/1805.04833.

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *IEEE Workshop on Automatic Speech Recognition and UnderstandingASRU*, pages 813–820.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation.

John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *24th Annual International ACM SIGIR Conference*, SIGIR '01.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Dongfang Li, Yifei Yu, Qingcai Chen, and Xinyu Li. 2019. Bertsel: Answer selection with pre-trained models. *CoRR*, abs/1905.07588.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *21st Annual International ACM SIGIR Conference*, SIGIR '98, pages 275–281. ACM.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, page 1913–1916.

Cicero dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *ACL*, pages 694–699.

Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *ACM SIGIR Conference*, pages 373–382.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2016. Lstm-based deep learning models for non-factoid answer selection. In *ICLR - Workshop Track*.

Yi Tay, Minh C. Phan, Luu Anh Tuan, and Siu Cheung Hui. 2017. Learning to rank question answer pairs with holographic dual lstm architecture. In *ACM SIGIR Conference*, pages 695–704.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Hyperbolic representation learning for fast and efficient neural question answering. In *International Conference on Web Search and Data Mining*, pages 583–591.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *CoRR*, abs/1702.03814.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Peng Xu, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Passage ranking with weak supervsion. *CoRR*, abs/1905.05910.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *EMNLP*, pages 2013–2018.

ChengXiang Zhai. 2008. Statistical language models for information retrieval a critical review. *Found. Trends Inf. Retr.*, 2(3):137–213.