Discriminative v.s. Generative Methods for Abstract Meaning Cloze-Style Machine Reading Comprehension

Haiying Huang
hhaiying1998@outlook.com

Ke Xu
kx@ucla.edu

Jingtong Kang
kjtabc123@g.ucla.edu

Abstract

We investigate the problem of abstract meaning cloze-style Machine Reading Comprehension. In this task, a machine is given a passage and a statement, and the machine is expected to fill abstract words in the blanks in the statement as a multiple-choice selection. We study two different approaches for solving this problem: 1) discriminative: one builds a model to predict the similarity between the passage and a completed statement, and selects the candidate which, after filled in, gives the highest similarity score; 2) generative: one builds a model that takes the passage and the incomplete statement as inputs and attempts to generate the missing word, and select the candidate that is most similar to the generated word. We evaluate and compare these two approaches on the SemEval-2021: Reading Comprehension of Abstract Meaning (ReCAM)(Zheng et al., 2021) and implement these approaches with different model architectures including LSTM, RoBERTa, and GPT.The result shows that for both discriminative and generative approaches, LSTM performs poorly, and ChatGPT performs relatively well. Generative "RoBERTa + fine tunning" and discriminative ChatGPT have the best accuracy, which might be our solution to this problem.

1 Problem Statement

In this project, we explore the problem of abstract meaning cloze-style Machine Reading Comprehension(MRC). Provided with a passage and a masked statement, the goal of the machine is to predict the missing abstract word from multiple-choice options. In contrast to research that focuses on the prediction of concrete concepts like name entities, our task faces the challenge of involving the models filling in abstract words, which do not pertain to specific tangible objects, events, or entities. This distinction requires the model to have an appropriate understanding of abstract terms that align with the overall meaning of the context.

Passage	... Observers have even named it after him, “Abenomics”. It is based on three key pillars of monetary policy to ensure long-term sustainable growth in the world’s third-largest economy, with fiscal stimulus and structural reforms. In this weekend’s upper house elections, ...
Question	Abenomics: The ***@placeholder*** and the risk.
Answer	(A) chance (B) prospective (C) government (D) objective (E) threat

Table 1: An Example of the task. The correct answer is **objective**.

The task is further divided into three subtasks based on two common definitions of abstractness, *imperceptibility* and *nonspecificity*, namely.

In the first definition of abstractness, in contrast to concrete objects and experiences such as water and swimming, abstract words refer to intangible ideas and concepts that cannot be directly perceived or sensed by humans

The other definition of abstractness is non-specific words(Theijssen et al., 2011), which represent general concepts or ideas. For example, *flower* is more abstract than *tulip* or *rose*. Subtask 2 focuses on the non-specificity in abstractness.

In Subtask 3, we evaluate the model’s generalizability across the two definitions of abstractness. This subtask includes two sets of experiments: 1) The models are trained on the training set of Subtask 1, and then tested on the test set of Subtask 2. 2) The models are trained on the training set of Subtask 2, and then tested on the test set of Subtask 1.

2 Background

2.1 Models

In this project, we implement three NLP models that represent important milestones in the field's progress.

LSTM LSTM is a recurrent neural network that handles vanishing gradients and captures long-term dependencies in sequential data.

RoBERTa Introduced by Google AI in 2018, RoBERTa(Liu et al., 2019) is a re-implementation of BERT with some hyperparameter modifications. It was built based on the masked language strategy of BERT. It removed BERT's objective of next-sentence prediction and utilized additional Web text corpus while BERT was pre-trained only on the English Wikipedia and Books corpus.

GPT GPT(Radford et al., 2019) is a large language model by OpenAI, trained on massive amounts of contextualized data using unsupervised pre-training, with billions of hyperparameters to learn complex relationships in the corpus.

2.2 State-of-the-art Approaches on MRC

On top of the pre-trained model, one team suggests an ensemble of ELECTRA-based models with task-adaptive pretraining and a multi-head attention multiple-choice classifier. (Jing et al.,2021) They achieved 95.11 and 94.89 on task1 and task2, respectively. They used BERT as their based models to train which give us the inspiration of using RoBERTa as our "Pretraining & Fine-tuning" method to compare and realized our discriminative and generative approach.

3 Our Method

3.1 Discriminative v.s Generative

The mainstream approach in solving cloze-style machine reading comprehension is the Discriminative and Generative approach. For the **Discriminative** approach, we substitute an option into the query. The problem then reduces to predict the similarity between the article and the option field query. It is ineffectively a binary classification task. We select the option that leads to the highest similarity at test time. For the **Generative** approach, we first concatenate the article and the query with placeholders and then train a model to generate the mask word in the query. We will select the option closest to the generated word at test time.

We want to compare the pros and cons of the two approaches. We hypothesize that the discriminative method can "peek" the correct option as its inputs so it utilizes more information and may have higher performance. The generative method is trained closer to a real-life setting so that it may have better generality.

3.2 Three NLP pipeline

We evaluate discriminative vs. generative approaches over three mainstream NLP pipelines:

1. Model engineering (e.g. LSTMs)
2. Pretraining & fine-tuning (e.g. RoBERTa)
3. Prompt engineering (e.g. ChatGPT)

4 Model Implementation

4.1 Siamese LSTM

In the discriminative approach, we applied Siamese LSTM, which has two LSTMs with shared weights that encoded the option filed questions and articles into latent vectors. We then obtained an absolute difference between the two latent vectors. Instead of the Manhattan distance in the original Siamese LSTM paper, we will apply a multilayer Neural Network to predict the binary similarity label from the difference of the latent vectors.

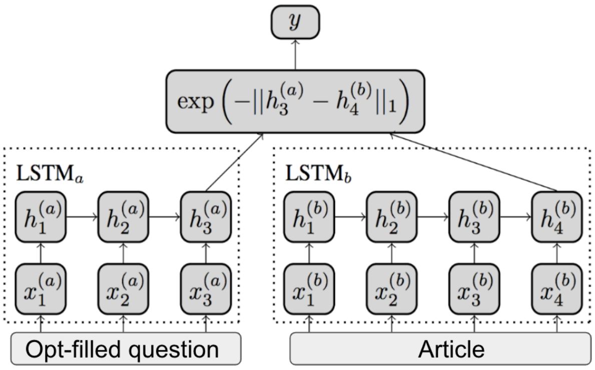

y
$\exp\left(-\|h_3^{(a)} - h_4^{(b)}\|_1\right)$
LSTM$_a$
LSTM$_b$
$h_1^{(a)}$ $h_2^{(a)}$ $h_3^{(a)}$
$x_1^{(a)}$ $x_2^{(a)}$ $x_3^{(a)}$
$h_1^{(b)}$ $h_2^{(b)}$ $h_3^{(b)}$ $h_4^{(b)}$
$x_1^{(b)}$ $x_2^{(b)}$ $x_3^{(b)}$ $x_4^{(b)}$
Opt-filled question
Article

Figure 1: Discriminative Siamese LSTM

For generative LSTM, we apply Siamese LSTM to encode the concatenated article question pair. The question is now a mask and uses MLP to transform the latent vector into a distribution over all vocabularies. We first trained only on cross-entropy loss, so only the positive word is considered. Then we use maximum likelihood to train the model. We get better results by promoting the positive word and penalizing the negative word.

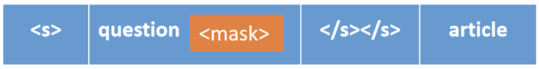


Figure 2: Input of generative Siamese LSTM

4.2 Fine-tuned RoBERTa

As mentioned in the project background, fed with a larger and more complicated test dataset while narrowing down the objectives, RoBERTa achieved better performance on the masked language modeling objective compared with BERT. Therefore, we adopted RoBERTa for this cloze-style comprehension task.

In the discriminative approach, each data sample is a concatenated query filled in by an option. Each statement is assigned a label of either True or False. We used Cross Entropy Loss for the discriminative approach. Therefore, a question with n options will be transformed into n concatenated queries. The model will choose the query with the highest True probability.

In the generative approach, for each input question, we will extract the probabilities of the options from the predicted probability distribution on the whole vocabulary. Then, the option with the highest probability is chosen as the answer to the question.

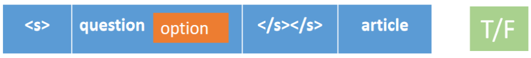


Figure 3: Input of Discriminative RoBERTa

4.3 ChatGPT

We design two kinds of prompts for ChatGPT to solve this problem. For the discriminative approach, we ask the chatbot to select one word from the five options to fill in the blank. The generative prompt consists of two rounds of Q&A. In the first question, we asked GPT to generate one word to replace the placeholder in the summary. In the second question, we ask GPT to select which option is most similar to the word chatbot generated.

We use ChatGPT API to process all the instances in the dataset automatically. We use model "gpt-3.5-turbo." We put a post request on the ChatGPT server, and the server will respond with an answer. Performance is good.

Discriminative Approach Prompt

[TASK] Read the passage and fill in the blank in the summary, using one of the options. Please only respond with the chosen option.
[PASSAGE]{article}
[SUMMARY]{question}
[OPTINS]{options}

Table 2: ChatGPT Dirscriminative Prompt

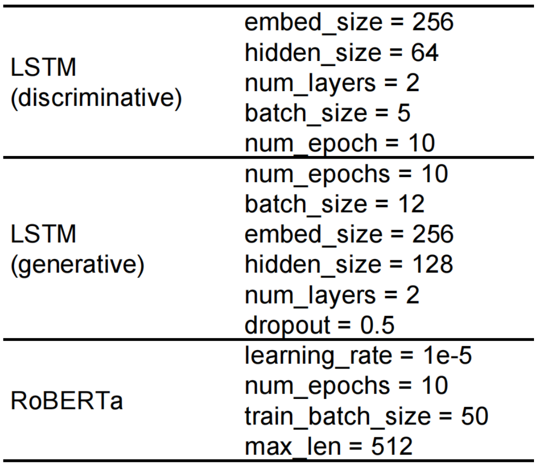
Generative Approach Prompt

[TASK] Read the passage and generate the placeholder in the summary. Please generate only one word.
[PASSAGE] {article}
[SUMMARY] {question}

[TASK] Which word from the options is most similar to the word **{answer}** in the following context? You must respond with one word from the options. [CONTEXT] {question filled with answer} [OPTIONS] {options}

Table 3: ChatGPT Generative Prompt

5 Results



LSTM (discriminative)	embed_size = 256
	hidden_size = 64
	num_layers = 2
	batch_size = 5
	num_epoch = 10

LSTM (generative)	num_epochs = 10
	batch_size = 12
	embed_size = 256
	hidden_size = 128
	num_layers = 2
	dropout = 0.5

RoBERTa	learning_rate = 1e-5
	num_epochs = 10
	train_batch_size = 50
	max_len = 512

Figure 4: Hyperparameter of the models

The dataset provided by SemEval-2021 (the original dataset) consists of 3,227 training samples and 837 validation samples on Subtask 1, and 3,318 training samples and 851 validation samples on Subtask 2.

Since the groundtruth label of the test data is not provided publicly by SemEval 2021, we combined the training data and validation data of the original data and split it into a new dataset. In the new dataset, train: validation: test = 7:1:2. The new dataset has 2,844 training samples, 407 validation samples, 813 test samples for Subtask 1. It has 2,918 training samples, 417 validation samples, and 834 test samples for Subtask2. After tuning the hyperparameters, we figure out the best parameters.

5.1 Subtask 1: Imperceptibility

Discriminative LSTM and ChatGPT perform better than its generative approach. RoBERTa generative approach performs better than the discriminative. Discriminative ChatGPT performs the best, followed by generative RoBERTa.

Model	Approach	Train Acc	Val Acc
LSTM	discriminate	0.6817	0.2676
	generative	0.8315	0.2306
RoBERTa	discriminate	0.8097	0.5221
	generative	0.9988	0.6392
GPT	discriminate	/	0.6891
	generative	/	0.6499

Table 4: Subtask 1 Results on Original Dataset

Model	Approach	Train Acc	Val Acc	Test Acc
LSTM	discriminate	0.6487	0.3145	0.3038
	generative	0.9364	0.2531	0.2386
RoBERTa	discriminate	0.7788	0.4914	0.4945
	generative	0.9821	0.6192	0.6138
GPT	discriminate	/	/	0.6543
	generative	/	/	0.6102

Table 5: Subtask 1 Results on New Dataset

5.2 Subtask 2: Non-Specificity

Generative LSTM and RoBERTa has better performance than discriminative. Discriminative ChatGPT have better performance. Generative RoBERTA has the best performance and the accuracy is 10% higher than discriminative. However, there is still severe overfitting, which contradicts our hypothesis that generalized model is closer to reality and might generalize better.

Model	Approach	Train Acc	Val Acc
LSTM	discriminate	0.8704	0.3000
	generative	0.8873	0.3114
RoBERTa	discriminate	0.8207	0.5922
	generative	0.9958	0.6874
GPT	discriminate	/	0.6801
	generative	/	0.6512

Table 6: Subtask 2 Results on Original Dataset

Model	Approach	Train Acc	Val Acc	Test Acc
LSTM	discriminate	0.8557	0.2998	0.2962
	generative	0.938	0.2950	0.3141
RoBERTa	discriminate	0.8273	0.6115	0.5540
	generative	0.9887	0.6307	0.6739
GPT	discriminate	/	/	0.6971
	generative	/	/	0.6559

Table 7: Subtask 2 Results on New Dataset

5.3 Subtask 3: Generalizability

All discriminative approach has better performance than generative ones. However, the test accuracy is lower than task1 & 2. This is reasonable because training and testing on specific task datasets is likely to perform better. Task 3 proves our hypothesis that discriminative performs better because it can peek at the correct options as the model input.

Model	Approach	Train on 1	Train on 2
		Test Acc on 2	Test Acc on 1
LSTM	discriminate	0.1939	0.2222
	generative	0.1751	0.1589
RoBERTa	discriminate	0.4712	0.4815
	generative	0.4771	0.4241

Table 8: Subtask 3 Results on Original Dataset

Model	Approach	Train on 1	Train on 2
		Test Acc on 2	Test Acc on 1
LSTM	discriminate	0.2122	0.2116
	generative	0.1811	0.1685
RoBERTa	discriminate	0.4880	0.4588
	generative	0.4868	0.4145

Table 9: Subtask 3 Results on New Dataset

5.4 Result Conclusion

The accuracy above refers to the percentage of correctly predicted questions. Among all subtasks, subtask 1 and subtask 2 have similar testing accuracy and are higher than subtask 3. The high performance of Generative RoBERTa on all subtasks reflects RoBERTa's strong ability in reading comprehension, so as ChatGPT API. All discriminative ChatGPT performs better than generative because the generative approach fails to give a clear single-word response. LSTM performs relatively poorly because given different options, the last-layer hidden representation of the opt-filled question is very similar since only one word in the LSTM input is different. LSTM is not expressive enough to capture this small semantic difference.

References

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

DL Theijssen, H van Halteren, LWJ Boves, and NHJ Oostdijk. 2011. On the difficulty of making concreteness concrete.

Boyuan Zheng, Xiaoyu Yang, Yu-Ping Ruan, Zhenhua Ling, Quan Liu, Si Wei, and Xiaodan Zhu. 2021. SemEval-2021 task 4: Reading comprehension of abstract meaning. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 37–50, Online. Association for Computational Linguistics.