Distractor Analysis and Selection for Multiple-Choice Cloze Questions for Second-Language Learners

Lingyu Gao[1], **Kevin Gimpel**[1], **Arnar Jensson**[2]
[1]Toyota Technological Institute at Chicago, [2]Cooori Japan
{lygao, kgimpel}@ttic.edu

Abstract

We consider the problem of automatically suggesting distractors for multiple-choice cloze questions designed for second-language learners. We describe the creation of a dataset including collecting manual annotations for distractor selection. We assess the relationship between the choices of the annotators and features based on distractors and the correct answers, both with and without the surrounding passage context in the cloze questions. Simple features of the distractor and correct answer correlate with the annotations, though we find substantial benefit to additionally using large-scale pretrained models to measure the fit of the distractor in the context. Based on these analyses, we propose and train models to automatically select distractors, and measure the importance of model components quantitatively.

1 Introduction

Multiple-choice cloze questions (MCQs) are widely used in examinations and exercises for language learners (Liang et al., 2018). The quality of MCQs depends not only on the question and choice of blank, but also on the choice of **distractors**, i.e., incorrect answers. Distractors, which could be phrases or single words, are incorrect answers that distract students from the correct ones.

According to Pho et al. (2014), distractors tend to be syntactically and semantically homogeneous with respect to the correct answers. Distractor selection may be done manually through expert curation or automatically using simple methods based on similarity and dissimilarity to the correct answer (Pino et al., 2008; Alsubait et al., 2014). Intuitively, optimal distractors should be sufficiently similar to the correct answers in order to challenge students, but not so similar as to make the question unanswerable (Yeung et al., 2019). However, past work usually lacks direct supervision for training, making it difficult to develop and evaluate automatic methods. To overcome this challenge, Liang et al. (2018) sample distractors as negative samples for the candidate pool in the training process, and Chen et al. (2015) sample questions and use manual annotation for evaluation.

In this paper, we build two datasets of MCQs for second-language learners with distractor selections annotated manually by human experts. Both datasets consist of instances with a sentence, a blank, the correct answer that fills the blank, and a set of candidate distractors. Each candidate distractor has a label indicating whether a human annotator selected it as a distractor for the instance. The first dataset, which we call MCDSENT, contains solely the sentence without any additional context, and the sentences are written such that they are understandable as standalone sentences. The second dataset, MCDPARA, contains sentences drawn from an existing passage and therefore also supplies the passage context.

To analyze the datasets, we design context-free features of the distractor and the correct answer, including length difference, embedding similarities, frequencies, and frequency rank differences. We also explore context-sensitive features, such as probabilities from large-scale pretrained models like BERT (Devlin et al., 2018). In looking at the annotations, we found that distractors are unchosen when they are either too easy or too hard (i.e., too good of a fit in the context). Consider the examples in Table 1. For the sentence "The large automobile **manufacturer** has a factory near here.", "beer" is too easy and "corporation" is too good of a fit, so both are rejected by annotators. We find that the BERT probabilities capture this tendency; that is, there is a nonlinear relationship between the distractor probability under BERT and the likelihood of annotator selection.

Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications, pages 102–114
July 10, 2020. ©2020 Association for Computational Linguistics

dataset	context with **correct answer**	distractor	label
MCDSent	How many people are planning to **attend** the party?	contribute	T
	The large automobile **manufacturer** has a factory near here.	beer	F
	The large automobile **manufacturer** has a factory near here.	corporation	F
	The large automobile **manufacturer** has a factory near here.	apartment	T
MCDPara	Stem cells are special cells that can divide to produce many different kinds of cells. When they divide, the new cells may be the same type of **cell** as the original cell....	plastic	F
	...These circumstances made it virtually impossible for salmon to mate. Therefore, the number of **salmon** declined dramatically.	thousands	T

Table 1: Example instances from MCDSent and MCDPara. Contexts are shown and correct answers are bold and underlined. Part of the paragraph contexts are replaced by ellipses.

We develop and train models for automatic distractor selection that combine simple features with representations from pretrained models like BERT and ELMo (Peters et al., 2018). Our results show that the pretrained models improve performance drastically over the feature-based models, leading to performance rivaling that of humans asked to perform the same task. By analyzing the models, we find that the pretrained models tend to give higher score to grammatically-correct distractors that are similar in terms of morphology and length to the correct answer, while differing sufficiently in semantics so as to avoid unaswerability.

2 Datasets

We define an **instance** as a tuple $\langle x, c, d, y \rangle$ where x is the **context**, a sentence or paragraph containing a blank; c is the **correct answer**, the word/phrase that correctly fills the blank; d is the **distractor candidate**, the distractor word/phrase being considered to fill the blank; and y is the **label**, a true/false value indicating whether a human annotator selected the distractor candidate.[1] We use the term **question** to refer to a set of instances with the same values for x and c.

2.1 Data Collection

We build two datasets with different lengths of context. The first, which we call MCDSent ("Multiple Choice Distractors with SENTence context"), uses only a single sentence of context. The second, MCDPara ("Multiple Choice Distractors with PARAgraph context"), has longer contexts (roughly one paragraph).

Our target audience is Japanese business people with TOEIC level 300-800, which translates to pre-intermediate to upper-intermediate level. Therefore, words from two frequency-based word lists, the New General Service List (NGSL; Browne et al., 2013) and the TOEIC Service List (TSL; Browne and Culligan, 2016), were used as a base for selecting words to serve as correct answers in instances. A proprietary procedure was used to create the sentences for both MCDSent and MCDPara tasks, and the paragraphs in MCDPara are excerpted from stories written to highlight the target words chosen as correct answers. The sentences are created following the rules below:

- A sentence must have a particular minimum and maximum number of characters.
- The other words in the sentence should be at an equal or easier NGSL frequency level compared with the correct answer.
- The sentence theme should be business-like.

All the MCDSent and MCDPara materials were created in-house by native speakers of English, most of whom hold a degree in Teaching English to Speakers of Other Languages (TESOL).

2.2 Distractor Annotation

We now describe the procedure used to propose distractors for each instance and collect annotations regarding their selection.

A software tool with a user interface was created to allow annotators to accept or reject distractor candidates. Distractor candidates are sorted automatically for presentation to annotators in order to favor those most likely to be selected. The distractor candidates are drawn from a proprietary dictionary, and those with the same part-of-speech (POS) as the correct answers (if POS data is available) are preferred. Moreover, the candidates that

[1] Each instance contains only a single distractor candidate because this matches our annotation collection scenario. Annotators were shown one distractor candidate at a time. The collection of simultaneous annotations of multiple distractor candidates is left to future work.

have greater similarity to the correct answers are preferred, such as being part of the same word learning section in the language learning course and the same NGSL word frequency bucket. There is also preference for candidates that have not yet been selected as distractors for other questions in the same task type and the same course unit.[2] After the headwords are decided through this procedure, a morphological analyzer is used to generate multiple inflected forms for each headword, which are provided to the annotators for annotation. Both the headwords and inflected forms are available when computing features and for use by our models.

Six annotators were involved in the annotation, all of whom are native speakers of English. Out of the six, four hold a degree in TESOL. Selecting distractors involved two-step human selection. An annotator would approve or reject distractor candidates suggested by the tool, and a different annotator, usually more senior, would review their selections. The annotation guidelines for MCDSent and MCDPara follow the same criteria. The annotators are asked to select distractors that are grammatically plausible, semantically implausible, and not obviously wrong based on the context. Annotators also must accept a minimum number of distractors depending on the number of times the correct answer appears in the course. Table 1 shows examples from MCDSent and MCDPara along with annotations.

2.3 Annotator Agreement

Some instances in the datasets have multiple annotations, allowing us to assess annotator agreement. We use the term "sample" to refer to a set of instances with the same x, c, and d. Table 2 shows the number of samples with agreement and disagreement for both datasets.[3] Samples with only one annotation dominate the data. Of the samples with multiple annotations, nearly all show agreement.

# anno.	MCDSent			MCDPara		
	agree	disagree	total	agree	disagree	total
1	-	-	232256	-	-	734063
2	2553	122	2675	9680	152	9841
3	121	2	123	493	3	496
4	17	0	17	62	0	62
5	10	0	10	12	0	12
6	0	0	0	2	0	2

Table 2: Numbers of samples for which annotators agree or disagree.

dataset	type	y	train	dev	test
MCDSent	questions	-	2,713	200	200
	instances	T	30,737	1,169	1,046
		F	191,908	6,420	6,813
MCDPara	questions	-	14,999	1,000	1,000
	instances	T	49,575	597	593
		F	688,804	7,620	8,364

Table 3: Dataset sizes in numbers of questions (a "question" is a set of instances with the same x and c) and instances, broken down by label (y) and data split.

2.4 Distractor Phrases

While most distractors are words, some are phrases, including 16% in MCDSent and 13% in MCDPara. In most cases, the phrases are constructed by a determiner or adverb ("more", "most", etc.) and another word, such as "most pleasant" or "more recently". However, some candidates show other patterns, such as noun phrases "South Pole", erroneously-inflected forms "come ed" and other phrases (e.g. "Promises Of", "No one").

2.5 Dataset Preparation

We randomly divided each dataset into train, development, and test sets. We remind the reader that we define a "question" as a set of instances with the same values for the context x and correct answer c, and in splitting the data we ensure that for a given question, all of its instances are placed into the same set. The dataset statistics are shown in Table 3. False labels are much more frequent than true labels, especially for MCDPara.

3 Features and Analysis

We now analyse the data by designing features and studying their relationships with the annotations.

3.1 Features

We now describe our features. The dataset contains both the headwords and inflected forms of both the correct answer c and each distractor candidate d. In defining the features below based on c and d for an instance, we consider separate features for the headword pair and the inflected form pair. For features that require embedding words, we use the 300-dimensional GloVe word embed-

[2] More specific details about this process are included in the supplementary material.

[3] We are unable to compute traditional inter-annotator agreement metrics like Cohen's kappa since we lack information about annotator identity for each annotation.

dings (Pennington et al., 2014) pretrained on the 42 billion token Common Crawl corpus. The GloVe embeddings are provided in decreasing order by frequency, and some features below use the line numbers of words in the GloVe embeddings, which correspond to frequency ranks. For words that are not in the GloVe vocabulary, their frequency ranks are $|N| + 1$, where N is the size of the GloVe vocabulary. We use the four features listed below:

- **length difference**: absolute value of length difference (in characters, including whitespace) between c and d.
- **embedding similarity**: cosine similarity of the embeddings of c and d. For phrases, we average the embeddings of the words in the phrase.
- **distractor frequency**: negative log frequency rank of d. For phrases, we take the max rank of the words (i.e., the rarest word is chosen).
- **freq. rank difference**: feature capturing frequency difference between c and d, i.e., $\log(1 + |r_c - r_d|)$ where r_w is the frequency rank of w.

3.2 Label-Specific Feature Histograms

Figure 1 shows histograms of feature values for each label.[4] Since the data is unbalanced, the histograms are "label-normalized", i.e., normalized so that the sum of heights for each label is 1. So, we can view each bar as the fraction of that label's instances with feature values in the given range.

The annotators favor candidates that have approximately the same length as the correct answers (Fig. 1, plot 1), as the true bars are much higher in the first bin (length difference 0 or 1). Selected distractors have moderate embedding similarity to the correct answers (Fig. 1, plot 2). If cosine similarity is very high or very low, then those distractors are much less likely to be selected. Such distractors are presumably too difficult or too easy, respectively.

Selected distractors are moderately frequent (Fig. 1, plot 3). Very frequent and very infrequent distractors are less likely to be selected. Distractors with small frequency rank differences (those on the left of plot 4) are more likely to be chosen (Fig. 1, plot 4). Large frequency differences tend to be found with very rare distractors, some of which may be erroneously-inflected forms.

We also computed Spearman correlations between feature values and labels, mapping the T/F labels to 1/0. Aside from what are shown in the feature histograms, we find that a distractor with a rare headword but more common inflected form is more likely to be selected, at least for MCDSENT. The supplementary material contains more detail on these correlations.

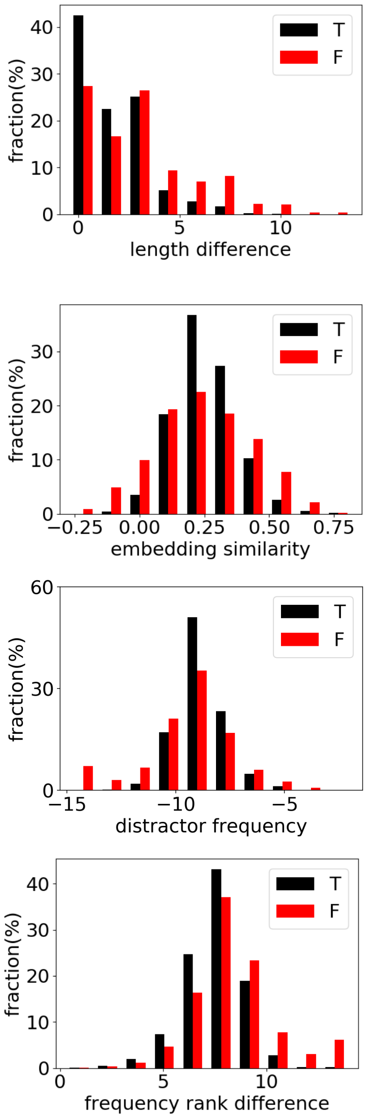


Figure 1: Label-specific feature histograms for MCDSENT.

3.3 Probabilities of Distractors in Context

We use BERT (Devlin et al., 2018) to compute probabilities of distractors and correct answers in the given contexts in MCDSENT. We insert a mask symbol in the blank position and compute the probability of the distractor or correct answer at that position.[5] Figure 2 shows histograms for correct answers and distractors (normalized by label). The

[4] We show plots here for the inflected form pairs; those for headword pairs are included in the supplementary material.

[5] For distractors with multiple tokens, we mask each position in turn and use the average of the probabilities.

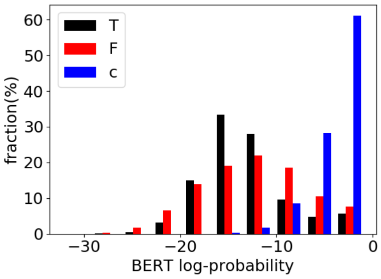


Figure 2: Histograms of BERT log-probabilities of selected distractors ("T"), unselected distractors ("F"), and correct answers ("*c*") in MCDSENT.

correct answers have very high probabilities. The distractor probabilities are more variable and the shapes of the histograms are roughly similar for the true and false labels. Interestingly, however, when the probability is very high or very low, the distractors tend to not be selected. The selected distractors tend to be located at the middle of the probability range. This pattern shows that BERT's distributions capture (at least partially) the nonlinear relationship between goodness of fit and suitability as distractors.

4 Models

Since the number of distractors selected for each instance is uncertain, our datasets could be naturally treated as a binary classification task for each distractor candidate. We now present models for the task of automatically predicting whether a distractor will be selected by an annotator. We approach the task as defining a predictor that produces a scalar score for a given distractor candidate. This score can be used for ranking distractors for a given question, and can also be turned into a binary classification using a threshold. We define three types of models, described in the subsections below.

4.1 Feature-Based Models

Using the features described in Section 3, we build a simple feed-forward neural network classifier that outputs a scalar score for classification. Only inflected forms of words are used for features without contexts, and all features are concatenated and used as the input of the classifier. For features that use BERT, we compute the log-probability of the distractor and the log of its rank in the distribution. For distractors that consist of multiple subword units, we mask each individually to compute the above features for each subword unit, then use the concatenation of mean, min, and max pooling of the features over the subword units. We refer to this model as M_{feat}.

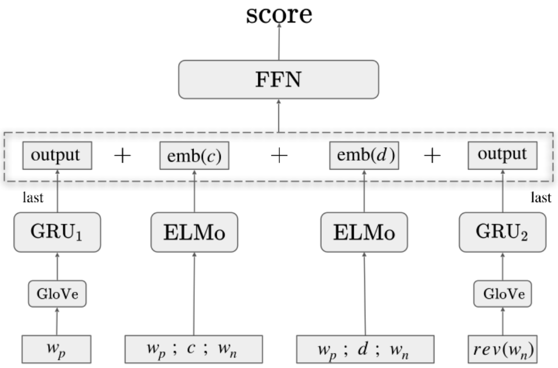


Figure 3: Illustration of the ELMo-based model M_{ELMo}, where semicolon refers to vector concatenation.

4.2 ELMo-Based Models

We now describe models that are based on ELMo (Peters et al., 2018) which we denote M_{ELMo}. Since MCDPARA instances contain paragraph context, which usually includes more than one sentence, we denote the model that uses the full context by $M_{ELMo}(\ell)$. By contrast, M_{ELMo} uses only a single sentence context for both MCDSENT and MCDPARA. We denote the correct answer by c, distractor candidate by d, the word sequence before the blank by w_p, and the word sequence after the blank by w_n, using the notation $rev(w_n)$ to indicate the reverse of the sequence w_n.

We use GloVe (Pennington et al., 2014) to obtain pretrained word embeddings for context words, then use two separate RNNs with gated recurrent units (GRUs; Cho et al., 2014) to output hidden vectors to represent w_p and w_n. We reverse w_n before passing it to its GRU, and we use the last hidden states of the GRUs as part of the classifier input. We also use ELMo to obtain contextualized word embeddings for correct answers and distractors in the given context, and concatenate them to the input. An illustration of this model is presented in Figure 3.

A feed-forward network (FFN) with 1 ReLU hidden layer is set on top of these features to get the score for classification:

$$FFN(z) = \max(0, zW_1 + b_1)W_2 + b_2$$

where z is a row vector representing the inputs shown in Figure 3. We train the model as a binary classifier by using a logistic sigmoid function on

dataset	precision		recall		F1	
	A	B	A	B	A	B
MCDSent	62.9	48.5	59.5	43.2	61.1	45.7
MCDPara	32.1	25.0	36.0	24.0	34.0	24.5

Table 4: Results of human performance on distractor selection for two human judges labeled A and B.

the output of $FFN(z)$ to compute the probability of the true label. We also experiment with the following variations:

- Concatenate the features from Section 3 with z.
- Concatenate the correct answer to the input of the GRUs on both sides (denoted *gru+c*).
- Concatenate the GloVe embeddings of the correct answers and distractors with z. We combine this with *gru+c*, denoting the combination *all*.

4.3 BERT-Based Models

Our final model type uses a structure similar to M_{ELMo} but using BERT in place of ELMo when producing contextualized embeddings, which we denote by M_{BERT} and $M_{BERT}(\ell)$ given different types of context. We also consider the variation of concatenating the features to the input to the classifier, i.e., the first variation described in Section 4.2. We omit the *gru+c* and *all* variations here because the BERT-based models are more computationally expensive than those that use ELMo.

5 Experiments

We now report the results of experiments with training models to select distractor candidates.

5.1 Evaluation Metrics

We use precision, recall, and F1 score as evaluation metrics. These require choosing a threshold for the score produced by our predictors. We also report the area under the precision-recall curve (AUPR), which is a single-number summary that does not require choosing a threshold.

5.2 Baselines

As the datasets are unbalanced (most distractor candidates are not selected), we report the results of baselines that always return "True" in the "baseline" rows of Tables 5 and 6. MCDSent has a higher percentage of true labels than MCDPara.

5.3 Estimates of Human Performance

We estimated human performance on the distractor selection task by obtaining annotations from NLP researchers who were not involved in the original data collection effort. We performed three rounds among two annotators, training them with some number of questions per round, showing the annotators the results after each round to let them calibrate their assessments, and then testing them using a final set of 30 questions, each of which has at most 10 distractors.

Human performance improved across rounds of training, leading to F1 scores in the range of 45-61% for MCDSent and 25-34% for MCDPara (Table 4). Some instances were very easy to reject, typically those that were erroneous word forms resulting from incorrect morphological inflection or those that were extremely similar in meaning to the correct answer. But distractors that were at neither extreme were very difficult to predict, as there is a certain amount of variability in the annotation of such cases. Nonetheless, we believe that the data has sufficient signal to train models to provide a score indicating suitability of candidates to serve as distractors.

5.4 Modeling and Training Settings

All models have one hidden layer for the feed-forward classifier. The M_{feat} classifier has 50 hidden units, and we train it for at most 30 epochs using Adam (Kingma and Ba, 2014) with learning rate 1e–3. We stop training if AUPR keeps decreasing for 5 epochs.[6] Although our primary metric of interest is AUPR, we also report optimal-threshold F1 scores on dev and test, tuning the threshold on the given set (so, on the test sets, the F1 scores we report are oracle F1 scores). The threshold is tuned within the range of 0.1 to 0.9 by step size 0.1.

For M_{ELMo} and $M_{ELMo}(\ell)$, we use ELMo (Original[7]) for the model, and BERT-large-cased to compute the BERT features from Section 3 (only applies to rows with "features = yes" in the tables). We increase the number of classifier hidden units to 1000 and run 20 epochs at most, also using Adam with learning rate $1e{-}3$. We stop training if AUPR does not improve for 3 epochs.

For M_{BERT} and $M_{BERT}(\ell)$, we applied the same training settings as M_{ELMo} and $M_{ELMo}(\ell)$. We com-

[6] We also tune by F1 score as another set of settings with similar trends, which are included in the supplementary material.

[7] https://allennlp.org/elmo

model	variant	development set				test set				BERT	features	best	threshold
		precision	recall	F1	AUPR	precision	recall	F1	AUPR			epoch	
baseline		15.4	100	26.7	-	13.3	100	23.5	-	-	-	-	-
M_{feat}		33.6	62.9	43.8	36.5	23.7	55.4	33.2	24.6	none	yes	28	0.2
		44.5	57.1	**50.0**	46.1	28.2	70.9	40.3	32.4	base	yes	25	0.2 (0.3)
		36.4	77.8	49.6	**47.0**	30.0	71.3	**42.2**	**34.5**	large	yes	22	0.2
M_{ELMo}	none	43.2	87.5	57.8	59.0	41.4	88.0	56.3	54.6	-	no	2	0.3
	gru+c	44.8	84.4	58.5	57.4	47.6	68.4	56.1	54.1	-	no	2	0.3 (0.4)
	all	47.2	88.9	61.7	61.2	48.3	75.0	58.7	55.8	-	no	2	0.3 (0.4)
	none	51.7	77.8	62.1	64.6	50.4	76.5	60.8	57.2	large	yes	3	0.3
	gru+c	55.7	73.3	63.3	65.3	49.1	82.3	61.5	**63.1**	large	yes	5	0.4 (0.3)
	all	56.2	74.4	**64.0**	**66.5**	49.8	80.8	**61.6**	58.8	large	yes	5	0.4 (0.3)
M_{BERT}		47.9	78.1	59.4	60.8	44.8	81.0	57.7	55.7	base	no	1	0.3
		49.6	79.3	61.0	64.1	45.3	80.2	57.9	53.4	large	no	1	0.3
		50.6	83.9	**63.2**	65.3	44.8	78.5	57.0	53.8	base	yes	12	0.1
		53.8	73.1	62.0	**66.5**	49.7	73.9	**59.4**	**56.3**	large	yes	2	0.4

Table 5: Results for MCDSENT. Boldface indicates the best F1/AUPR on dev/test for each model type. We include the threshold tuned on the test set in parentheses when it differs from the threshold tuned on dev.

model	development set				test set				BERT	features	best epoch	threshold
	precision	recall	F1	AUPR	precision	recall	F1	AUPR				
baseline	7.3	100	13.5	-	6.6	100	12.4	-	-	-	-	-
M_{feat}	15.3	63.1	24.6	17.3	14.5	63.6	23.6	15.5	-	yes	23	0.1
	18.2	69.2	28.9	21.6	16.3	65.6	26.1	**19.1**	base	yes	27	0.1
	19.8	64.0	**30.2**	**22.3**	16.9	64.2	**26.8**	18.8	large	yes	22	0.1
M_{ELMo}	35.4	47.7	40.7	38.4	26.1	75.6	38.8	30.4	-	no	5	0.3 (0.2)
	37.9	61.3	**46.9**	**46.8**	34.6	63.9	44.9	37.6	large	yes	7	0.3
$M_{ELMo}(\ell)$	30.5	61.1	40.7	36.6	29.1	61.6	39.5	33.2	-	no	5	0.3
	37.1	62.7	46.6	43.7	34.4	65.1	**45.0**	**40.1**	large	yes	6	0.3
M_{BERT}	35.4	61.6	45.0	40.9	29.2	58.7	39.0	30.1	base	no	2	0.2
	33.0	63.7	43.5	40.9	29.1	65.1	40.2	32.4	large	no	2	0.2
	44.3	55.4	**49.3**	**47.3**	31.5	73.2	**44.0**	36.7	base	yes	2	0.3 (0.2)
	35.6	66.0	46.2	45.0	35.5	54.5	43.0	36.6	large	yes	2	0.2 (0.3)
$M_{BERT}(\ell)$	33.1	65.3	43.9	39.7	28.8	66.4	40.2	29.8	base	no	2	0.2
	37.4	67.3	48.1	46.0	31.3	69.1	43.1	**37.0**	base	yes	2	0.2

Table 6: Results for MCDPARA.

pare the BERT-base-cased and BERT-large-cased variants of BERT. When doing so, the BERT features from Section 3 use the same BERT variant as that used for contextualized word embeddings.

For all models based on pretrained models, we keep the parameters of the pretrained models fixed. However, we do a weighted summation of the 3 layers of ELMo, and all layers of BERT except for the first layer, where the weights are trained during the training process.

5.5 Results

We present our main results for MCDSENT in Table 5 and for MCDPARA in Table 6.

Feature-based models. The feature-based model, shown as M_{feat} in the upper parts of the tables, is much better than the trivial baseline. Including the BERT features in M_{feat} improves performance greatly (10 points in AUPR for MCDSENT), showing the value of using the context effectively with a powerful pretrained model. There is not a large difference between using BERT-base and BERT-large when computing these features.

ELMo-based models. Even without features, M_{ELMo} outperforms M_{feat} by a wide margin. Adding features to M_{ELMo} further improves F1 by 2-5% for MCDSENT and 5-6% for MCDPARA. The F1 score for M_{ELMo} on MCDSENT is close to human performance, and on MCDPARA the F1 score outperforms humans (see Table 4). For

MCDSENT, we also experiment with using the correct answer as input to the context GRUs (*gru+c*), and additionally concatenating the GloVe embeddings of the correct answers and distractors to the input of the classifier (*all*). Both changes improve F1 on dev, but on test the results are more mixed.

BERT-based models. For M_{BERT}, using BERT-base is sufficient to obtain strong results on this task and is also cheaper computationally than BERT-large. Although M_{BERT} with BERT-base has higher AUPR on dev, its test performance is close to M_{ELMo}. Adding features improves performance for MCDPARA (3-5% F1), but less than the improvement found for M_{ELMo}. While M_{feat} is aided greatly when including BERT features, the features have limited impact on M_{BERT}, presumably because it already incorporates BERT in its model.

Long-context models. We now discuss results for the models that use the full context in MCDPARA, i.e., $M_{ELMo}(\ell)$ and $M_{BERT}(\ell)$. On dev, M_{ELMo} and M_{BERT} outperform $M_{ELMo}(\ell)$ and $M_{BERT}(\ell)$ respectively, which suggests that the extra context for MCDPARA is not helpful. However, the test AUPR results are better when using the longer context, suggesting that the extra context may be helpful for generalization. Nonetheless, the overall differences are small, suggesting that either the longer context is not important for this task or that our way of encoding the context is not helpful. The judges in our manual study (Sec. 5.3) rarely found the longer context helpful for the task, pointing toward the former possibility.

5.6 Statistical Significance Tests

For better comparison of these models' performances, a paired bootstrap resampling method is applied (Koehn, 2004). We repeatedly sample with replacement 1000 times from the original test set with sample size equal to the corresponding test set size, and compare the F1 scores of two models. We use the thresholds tuned by the development set for F1 score computations, and assume significance at a p value of 0.05.

- For M_{ELMo}, $M_{ELMo}(\ell)$, M_{BERT} and $M_{BERT}(\ell)$, the models with features are significantly better than their feature-less counterparts ($p < 0.01$).[8]
- When both models use features, $M_{ELMo}(\ell)$ is almost the same as M_{ELMo} ($p = 0.477$). However, when both do not use features, $M_{ELMo}(\ell)$ is significantly better ($p < 0.01$).
- When using BERT-base-cased, $M_{BERT}(\ell)$ is better than M_{BERT}, but not significantly so ($p = 0.4$ with features and 0.173 without features).
- On MCDPARA, switching from BERT-base to BERT-large does not lead to a significant difference for M_{BERT} without features (BERT-large is better with $p = 0.194$) or M_{BERT} with features (BERT-base is better with $p = 0.504$). For MCDSENT, M_{BERT} with BERT-large is better both with and without features ($p < 0.2$).
- On MCDPARA, $M_{BERT}(\ell)$ outperforms $M_{ELMo}(\ell)$ without features but not significantly. With features, $M_{ELMo}(\ell)$ is better with $p = 0.052$.
- On MCDSENT, M_{BERT} without features (BERT-large-cased) is better than M_{ELMo} without features, but not significantly so ($p = 0.386$). However, if we add features or use M_{BERT} with BERT-base-cased, M_{ELMo} is significantly better ($p < 0.01$).
- On MCDPARA, M_{ELMo} is nearly significantly better than M_{BERT} when both use features ($p = 0.062$). However, dropping the features for both models makes M_{BERT} significantly outperform M_{ELMo} ($p = 0.044$).

5.7 Examples

Figure 4 shows an example question from MCDSENT, i.e., "The bank will **notify** its customers of the new policy", and two subsets of its distractors. The first subset consists of the top seven distractors using scores from M_{ELMo} with features, and the second contains distractors further down in the ranked list. For each model, we normalize its distractor scores with min-max normalization.[9]

Overall, model rankings are similar across models, with all distractors in the first set ranked higher than those in the second set. The high-ranking but unselected distractors ("spell", "consult", and "quit") are likely to be reasonable distractors for second-language learners, even though they were not selected by annotators.

We could observe the clustering of distractor ranks with similar morphological inflected form in some cases, which may indicate that the model makes use of the grammatical knowledge of pretrained models.

[8] We only use BERT-base-cased for $M_{BERT}(\ell)$ due to computational considerations.

[9] Given original data x, we use $(x - \min(x))/(\max(x) - \min(x))$ to normalize it.

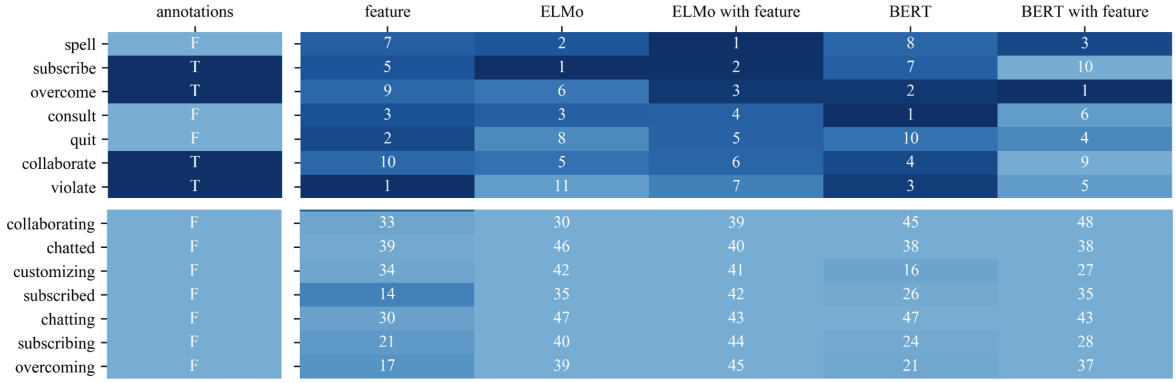


Figure 4: Ranks of distractors for question "The bank will **notify** its customers of the new policy." The colors represent the normalized scores of the models and the numbers in the cells are the ranks of the candidates.

6 Related Work

Existing approaches to distractor selection use WordNet (Fellbaum, 1998) metrics (Mitkov and Ha, 2003; Chen et al., 2015), word embedding similarities (Jiang and Lee, 2017), thesauruses (Sumita et al., 2005; Smith et al., 2010), and phonetic and morphological similarities (Pino and Eskenazi, 2009). Other approaches consider grammatical correctness, and introduce structural similarities in an ontology (Stasaski and Hearst, 2017), and syntactic similarities (Chen et al., 2006). When using broader context, bigram or n-gram co-occurrence (Susanti et al., 2018; Hill and Simha, 2016), context similarity (Pino et al., 2008), and context sensitive inference (Zesch and Melamud, 2014) have also been applied to distractor selection.

Based on these heuristic features, Liang et al. (2018) assemble these features and apply neural networks, training the model to predict the answers within a lot of candidates. Yeung et al. (2019) further applies BERT for ranking distractors by masking the target word. As we have two manually annotated datasets that have different lengths of contexts, we adopt both word pair features and the context-specific distractor probabilities to build our feature-based models. Moreover, we build both ELMo-based and BERT-based models, combining them with our features and measuring the impact of these choices on performance.

7 Conclusion

We described two datasets with annotations of distractor selection for multiple-choice cloze questions for second-language learners. We designed features and developed models based on pretrained language models. Our results show that the task is challenging for humans and that the strongest models are able to approach or exceed human performance. The rankings of distractors provided by our models appear reasonable and can reduce a great deal of human burden in distractor selection. Future work will use our models to collect additional training data which can then be refined in a second pass by limited human annotation. Other future work can explore the utility of features derived from pretrained question answering models in scoring distractors.

References

Tahani Alsubait, Bijan Parsia, and Uli Sattler. 2014. Generating multiple choice questions from ontologies: Lessons learnt. In *OWLED*, pages 73–84. Citeseer.

C. Browne and B. Culligan. 2016. The TOEIC Service List. http://www.newgeneralservicelist.org.

C. Browne, B. Culligan, and J. Phillips. 2013. The New General Service List. http://www.newgeneralservicelist.org.

Chia-Yin Chen, Hsien-Chin Liou, and Jason S Chang. 2006. Fast–an automatic generation system for grammar tests. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 1–4.

Tao Chen, Naijia Zheng, Yue Zhao, Muthu Kumar Chandrasekaran, and Min-Yen Kan. 2015. Interactive second language learning from news websites. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pages 34–42.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

Jennifer Hill and Rahul Simha. 2016. Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and google n-grams. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 23–30.

Shu Jiang and John Lee. 2017. Distractor generation for chinese fill-in-the-blank items. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C Lee Giles. 2018. Distractor generation for multiple choice questions using learning to rank. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 284–290.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*, pages 17–22. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Van-Minh Pho, Thibault André, Anne-Laure Ligozat, Brigitte Grau, Gabriel Illouz, and Thomas François. 2014. Multiple choice question corpus analysis for distractor characterization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Juan Pino and Maxine Eskenazi. 2009. Semi-automatic generation of cloze question distractors effect of students' l1. In *International Workshop on Speech and Language Technology in Education*.

Juan Pino, Michael Heilman, and Maxine Eskenazi. 2008. A selection strategy to improve cloze question quality. In *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada*, pages 22–32.

Simon Smith, PVS Avinesh, and Adam Kilgarriff. 2010. Gap-fill tests for language learners: Corpus-driven item generation. In *Proceedings of ICON-2010: 8th International Conference on Natural Language Processing*, pages 1–6. Macmillan Publishers.

Katherine Stasaski and Marti A Hearst. 2017. Multiple choice question generation utilizing an ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 303–312.

Eiichiro Sumita, Fumiaki Sugaya, and Seiichi Yamamoto. 2005. Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 61–68.

Yuni Susanti, Takenobu Tokunaga, Hitoshi Nishikawa, and Hiroyuki Obari. 2018. Automatic distractor generation for multiple-choice english vocabulary questions. *Research and Practice in Technology Enhanced Learning*, 13(1):15.

Chak Yan Yeung, John SY Lee, and Benjamin K Tsou. 2019. Difficulty-aware distractor generation for gap-fill items. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 159–164.

Torsten Zesch and Oren Melamud. 2014. Automatic generation of challenging distractors using context-sensitive inference rules. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148.

A Supplemental Material

A.1 Dataset

There are some problematic words in the dataset, such as 'testing, test', 'find s', 'find ed' in MCDSent/MCDPara candidate words. There are also some extra spaces (or non-breaking spaces) at the start or end of words. To keep the words the same as what the annotators saw, we only remove leading/trailing white space, and replace non-breaking spaces with ordinary spaces. By comparing the percentages of the circumstances where spaces are included in the string before/after tokenization, we find the percentage of extra spaces presented in Table 7. The vocabulary size after tokenization is presented in Table 8.

%	headword(c)	c	headword(d)	d
MCDSent	0	0	0.0168	0.0332
MCDPara	0.0160	0.0307	0.0364	0.0622

Table 7: Percentage of extra spaces (excluding those that are in the middle of words), where headword(c) denotes headword of correct answer, and d denotes distractor candidates of inflected forms. .

	headword(c)	c	headword(d)	d
MCDSent	2571	2731	3514	11423
MCDPara	2683	4174	3582	13749

Table 8: Vocabulary sizes.

A.2 Distractor Annotation

The software tool suggested distractor candidates based on the following priority ranking:

1. It is in a proprietary dictionary.
2. It has the same part-of-speech (POS) as the correct answer (if POS data is available) and satisfies 1.
3. It is part of a proprietary learnable word list for the language learning course under consideration, and satisfies 2.
4. It is in the same course as the correct answer and satisfies 3.
5. It is in the same proprietary study material bundle as the correct answer and satisfies 4.
6. It is in the previous or same study material as the correct answer and satisfies 5.
7. It is in the same study material as the correct answer and satisfies 6.
8. It is in the same NGSL frequency word list band as the correct answer and satisfies 7.
9. It is not used as a distractor for another word with the same task type in the same material at the time that the distractor list for quality assurance (QA) is loaded, and satisfies 8.

A.3 Context Position

Sometimes the blank resides at the start or end of the context, counts of which are shown in Table 9. The percentage when there is only one sentence as context in MCDPara is 0.894%.

%	sent start	sent end	para start	para end
FB1	3.058	0.005	-	-
FB3	2.640	0.342	18.272	22.165

Table 9: Position of the candidates, where "sent" denotes sentence and "para" denotes paragraph. "para start" mean that the sentence containing the blank is at the beginning of the paragraph.

A.4 Correlations of Features and Annotations

The Spearman correlations for these features are presented in Table 10. The overall correlations are mostly close to zero, so we explore how the relationships vary for different ranges of feature values below. Nonetheless, we can make certain observations about the correlations:

- Length difference has a weak negative correlation with annotations, which implies that the probability of a candidate being selected decreases when the absolute value of word length difference between the candidate and correct answer increases. The same conclusion can be drawn with headword pairs although the correlation is weaker.
- Embedding similarity has a very weak correlation (even perhaps none) with the annotations. However, the correlation for headwords is slightly negative while that for inflected forms is slightly positive, suggesting that annotators tend to select distractors with different lemmas than the correct answer, but similar inflected forms.
- Candidate frequency also has a very weak correlation with annotations (negative for headwords and positive for inflected forms). Since the feature is the negative log frequency rank, a distractor with a rare headword but more common inflected form is more likely to be selected, at least for MCDSent.

feature	MCDSent head	MCDSent infl	MCDPara head	MCDPara infl
length difference	-0.116	-0.171	-0.145	-0.173
embedding similarity	-0.018	0.026	-0.014	0.016
candidate frequency	-0.057	0.113	-0.062	0.028
freq. rank difference	-0.048	-0.161	-0.033	-0.091

Table 10: Spearman correlations with T/F choices, where "head" denotes headword pairs, and "infl" denotes inflected form pairs.

- Frequency rank difference has a weak negative correlation with annotations, and this trend is more significant with the inflected form pair. This implies that annotators tend to select distractors in the same frequency range as the correct answers.

The correlations are not very large in absolute terms, however we found that there were stronger relationships for particular ranges of these feature values and we explore this in the next section.

A.5 Label-Specific Feature Histograms

Figure 5 shows histograms of the feature values for each label on headword pairs.

A.6 Results Tuned Based on F1

We report our results tuned based on F1 in Table 11 and 12.

A.7 Supplement for Analysis

The example for MCDPara is as below, and two sets of its distractors are shown in Figure 6.

- MCDPara: A few years have passed since the Great Tohoku Earthquake occurred. It has been extremely costly to rebuild the damaged areas from scratch, with well over $200 billion dollars provided for reconstruction. However, the **availability** of these funds has been limited. However, a large portion of the money has been kept away from the victims due to a system which favors construction companies....

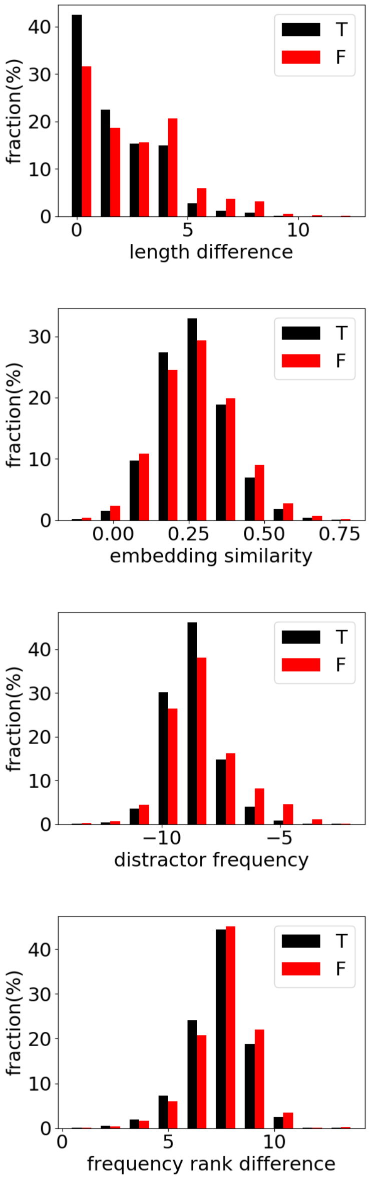


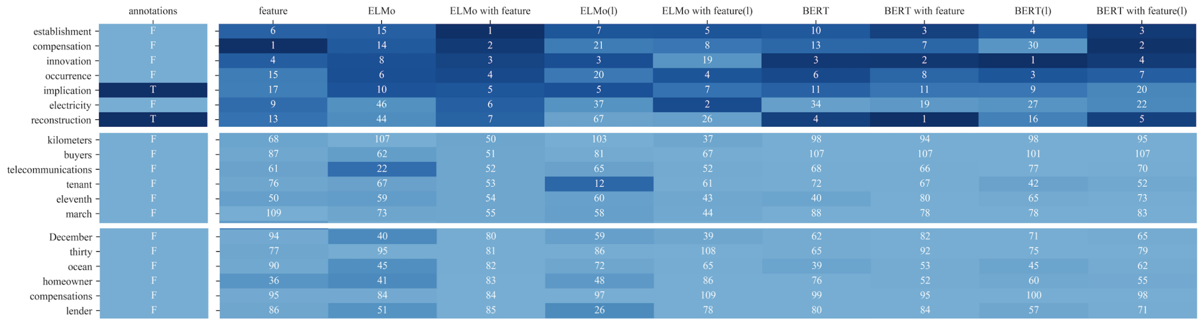
Figure 5: Label-normalized feature histograms for MCDSent (headword pairs).

model	variant	development set precision	 recall	 F1	 AUPR	test set precision	 recall	 F1	 AUPR	BERT	features	best epoch	threshold
baseline		15.4	100	26.7	-	13.3	100	23.5	-	-	-	-	-
M_{feat}		33.3	64.8	44.0	35.1	23.2	59.1	33.3	25.0	none	yes	26	0.2
		42.1	67.0	**51.7**	45.4	31.5	57.4	40.7	32.3	base	yes	26	0.2
		41.3	67.1	51.1	**46.7**	32.4	56.6	**41.2**	**33.9**	large	yes	25	0.3
M_{ELMo}	none	49.0	79.1	60.5	58.5	46.5	75.7	57.6	53.9	-	no	6	0.3
	gru+c	49.7	77.6	60.6	54.1	46.1	73.3	56.7	53.5	-	no	3	0.4
	all	52.9	75.8	62.3	60.4	48.0	75.9	58.8	57.6	-	no	2	0.4
	none	51.0	84.0	63.4	63.1	47.7	81.4	60.1	60.6	large	yes	3	0.3
	gru+c	56.9	72.3	63.7	59.1	50.6	75.9	**60.8**	58.6	large	yes	5	0.4
	all	53.5	80.8	**64.4**	**63.4**	50.8	75.5	**60.8**	**59.6**	large	yes	3	0.4
M_{BERT}		48.8	85.5	62.1	56.6	43.8	82.8	57.3	51.5	base	no	4	0.2
		49.6	80.8	61.5	59.1	45.2	79.7	57.7	54.9	large	no	3	0.3
		51.5	84.2	**63.9**	61.7	46.0	78.6	58.0	55.0	base	yes	6	0.2
		51.4	81.1	62.9	**64.7**	46.4	79.8	**58.7**	**57.5**	large	yes	6	0.2

Table 11: Results for MCDSent tuned based on F1.

model	development set precision	 recall	 F1	 AUPR	test set precision	 recall	 F1	 AUPR	BERT	features	best epoch	threshold
baseline	7.3	100	13.5	-	6.6	100	12.4	-	-	-	-	-
M_{feat}	17.1	53.1	25.9	15.9	15.6	51.3	23.9	15.0	-	yes	14	0.1
	19.5	63.0	29.8	20.4	17.6	61.0	22.3	**18.6**	base	yes	22	0.1
	20.4	63.1	**30.8**	**22.3**	16.7	62.7	**26.4**	**18.6**	large	yes	25	0.1
M_{ELMo}	35.2	55.4	43.1	37.0	31.2	54.6	39.8	33.9	-	no	5	0.3
	40.2	61.3	**48.5**	**43.8**	34.1	59.4	**43.3**	35.2	large	yes	5	0.3
$M_{ELMo}(\ell)$	28.7	72.9	41.2	33.8	25.7	71.3	37.7	30.3	-	no	2	0.2
	36.2	67.3	47.1	40.8	31.0	65.6	42.1	**37.3**	large	yes	7	0.3
M_{BERT}	35.8	64.2	46.0	39.3	28.9	64.3	39.9	34.5	base	no	5	0.2
	35.2	62.1	45.0	38.3	26.9	60.5	37.3	29.3	large	no	6	0.1
	44.3	55.4	49.3	**47.3**	34.6	56.2	42.8	36.7	base	yes	2	0.3
	37.8	63.3	47.4	44.0	32.7	66.1	**43.7**	**38.1**	large	yes	3	0.2
$M_{BERT}(\ell)$	34.0	64.3	44.5	36.7	29.6	62.1	40.1	32.1	base	no	5	0.2
	43.3	57.5	**49.4**	45.4	33.3	60.9	43.1	35.8	base	yes	3	0.3

Table 12: Results for MCDPara tuned based on F1.



	annotations	feature	ELMo	ELMo with feature	ELMo(l)	ELMo with feature(l)	BERT	BERT with feature	BERT(l)	BERT with feature(l)
establishment	F	6	15	1	7	5	10	3	4	3
compensation	F	1	14	2	21	8	13	7	30	2
innovation	F	4	8	3	3	19	3	2	1	4
occurrence	F	15	6	4	20	4	6	8	3	7
implication	T	17	10	5	5	7	11	11	9	20
electricity	F	9	46	6	37	2	34	19	27	22
reconstruction	T	13	44	7	67	26	4	1	16	5
kilometers	F	68	107	50	103	37	98	94	98	95
buyers	F	87	62	51	81	67	107	107	101	107
telecommunications	F	61	22	52	65	52	68	66	77	70
tenant	F	76	67	53	12	61	72	67	42	52
eleventh	F	50	59	54	60	43	40	80	65	73
march	F	109	73	55	58	44	88	78	78	83
December	F	94	40	80	59	39	62	82	71	65
thirty	F	77	95	81	86	108	65	92	75	79
ocean	F	90	45	82	72	65	39	53	45	62
homeowner	F	36	41	83	48	86	76	52	60	55
compensations	F	95	84	84	97	109	99	95	100	98
lender	F	86	51	85	26	78	80	84	57	71

Figure 6: Ranks for distractor candidates of MCDPara question “However, the **availability** of these funds has been limited.” along with annotations.