Automatic Distractor Suggestion for Multiple-Choice Tests Using Concept Embeddings and Information Retrieval

Le An Ha and Victoria Yaneva
Research Institute in Information and Language Processing,
University of Wolverhampton, UK
ha.l.a@wlv.ac.uk, v.yaneva@wlv.ac.uk

Abstract

Developing plausible distractors (wrong answer options) when writing multiple-choice questions has been described as one of the most challenging and time-consuming parts of the item-writing process. In this paper we propose a fully automatic method for generating distractor suggestions for multiple-choice questions used in high-stakes medical exams. The system uses a question stem and the correct answer as an input and produces a list of suggested distractors ranked based on their similarity to the stem and the correct answer. To do this we use a novel approach of combining concept embeddings with information retrieval methods. We frame the evaluation as a prediction task where we aim to "predict" the human-produced distractors used in large sets of medical questions, i.e. if a distractor generated by our system is good enough it is likely to feature among the list of distractors produced by the human item-writers. The results reveal that combining concept embeddings with information retrieval approaches significantly improves the generation of plausible distractors and enables us to match around 1 in 5 of the human-produced distractors. The approach proposed in this paper is generalisable to all scenarios where the distractors refer to concepts.

1 Introduction

Multiple-choice tests are one of the most widely used forms of both formative and summative assessment and are a probably the most prominent feature of high-stakes standardized exams (Gierl et al., 2017). Administering such exams requires the development of a large number of good-quality multiple-choice questions (MCQs). To illustrate the need to have a large number of questions, Breithaupt et al. (2009) report that a 40-item computer adaptive test for high-stakes examination administered twice a year would require a bank with 2,000 items and Gierl et al. (2017) estimate that the cost of developing an item bank of this size would be between 3,000,000 and 5,000,000 USD. Naturally, this creates the incentive to automate the test production as much as possible and has resulted in a large number of papers on the topic of automatic MCQ generation.

An important aspect of MCQ development is the generation of plausible distractors (wrong answer options), as they can help control for the difficulty of the item, reduce random guessing and discriminate properly between different levels of student ability (Alsubait et al., 2013). This task poses a challenge to both humans and machines and is especially demanding in the field of medical exams. For example, an analysis of 514 human-produced items including 2056 options (1542 distractors and 514 correct responses), administered to undergraduate nursing students, indicated that "Only 52.2% (n = 805) of all distractors were functioning effectively and 10.2% (n = 158) had a choice frequency of 0." (Tarrant et al., 2009). Items with more functioning distractors were found to be more difficult and more discriminating.

A particular challenge for the automatic development of MCQ distractors for the medical domain is the coverage of the ontologies, which could be too narrow in some cases, and too broad in others, and the need to rank the candidates in order to select the best ones. At the same time, this domain is of particular need of automated assistance, as the requirement for a very specialized knowledge makes the recruitment of item-writers and the test development procedure even more costly.

To address this issue we propose a method to fully automatically suggest distractors for MCQs given a stem[1] and a correct answer. The data used

[1] In this study we refer to the following components of an MCQ. The *stem* denotes the part that identifies the question or problem; *answer options* refer to all possible answers that an

Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 389–398
New Orleans, Louisiana, June 5, 2018.

in this study features two sets of 1,441 MCQs and 369 MCQs from the United States Medical Licensing Examination (USMLE) for which we have the stem, all answer options and information on which the correct answer is. We compare two approaches to suggesting distractors based on: i) concept embeddings only and ii) concept embeddings reranked using information retrieval techniques. The evaluation of these approaches is formulated as a prediction task, where each system uses the stem and the correct answer as an input and tries to predict the existing distractor options for each item as an output. The contributions of this study are as follows:

- We propose a novel method for distractor generation and selection based on concept embeddings reranked using information retrieval, which can successfully suggest relevant distractors given an item stem and the correct answer option.
- We show that the ranking based on information-retrieval methods improves the distractor prediction significantly.
- The approach used in this study is generalisable to all scenarios where the answer options refer to concepts. Furthermore, it can generate distractors for any item given that the correct answer features as an entry in the ontology, as opposed to only items generated by a specific method.

The rest of this paper is organised as follows. The next section presents related work on automatic item generation with special emphasis on distractor generation and evaluation. Section 3 describes the data sets used in this study and Section 4 describes our method. The results are reported in Section 6, discussed in Section 6 and summarised in Section 7.

2 Related Work

The automatic generation of multiple-choice questions (MCQs) has received a lot of attention in the past two decades, offering a range of approaches such as template-based item generation (Gierl et al., 2015, 2016; Lai et al., 2016), ontology-based item generation (Holohan et al., 2006; Papasalouros et al., 2008; Alsubait et al., 2014), and generation of items from unstructured text (Mitkov and Ha, 2003; Brown et al., 2005; Heilman, 2011; Hoshino and Nakagawa, 2005; Majumder and Saha, 2015).

The work most relevant to the field of MCQ generation for the medical education domain relies on a semi-automatic approach for template-based language generation, where variations of items are produced based on an item template (Gierl et al., 2016; Lai et al., 2016). An item template is a model that highlights the features which can be manipulated in order to generate a variation of the MCQ (e.g. strings and numerals) and thus increase the item bank for an exam. The method is semi-automatic in that it requires content developers to specify the initial item template and the information which could potentially be varied. For numeric options, the distractors are generated based on a pre-defined formula for each distractor candidate. For key feature options, the distractors may be from the same category as the correct answer, such as the same concept, topic, or idea at varying hyponymic or hypernymic levels. Evaluation of 13 MCQs generated in this way by 455 Canadian and international medical graduates revealed that the generated items were consistently discriminative in measuring the different levels of abilities of the students (Lai et al., 2016).

In terms of automatic distractor generation, systems which generate MCQs based on unstructured text have a limited ability to infer implicit relations within the text and generate plausible distractors (Alsubait et al., 2013). However, Mitkov and Ha (2003) select distractors by using WordNet to compute concepts semantically close to the correct answer by retrieving hypernyms, hyponyms, and coordinates of the term. In the event of WordNet returning too many concepts, preference is given to those appearing in the corpus and in the event that no concepts are returned the corpus is searched for noun phrases with the same head which are then used as distractors. Evaluation of 24 MCQs with test-takers revealed that the distractors were able to discriminate between high and low-ability students, where only 3 distractors were selected by no student and 6 were classed as *poor*, for misleading high-ability students.

Finally, most ontology-based MCQ generation systems output distractors based on hierarchical parent and sibling relations between the correct answer and the candidates (Papasalouros et al.,

examinee can choose from; *distractors* are the wrong answer options, and the *correct answer* is the correct answer option. Please refer to Table 1 for an example of a MCQ item.

An example of an item from the public data set
A 55-year-old woman with small cell carcinoma of the lung is admitted to the hospital to undergo chemotherapy. Six days after treatment is started, she develops a temperature of 38C (100.4F). Physical examination shows no other abnormalities. Laboratory studies show a leukocyte count of 100/mm3 (5% segmented neutrophils and 95% lymphocytes).
Which of the following is the most appropriate pharmacotherapy to increase this patient's leukocyte count?
(A) Darbepoetin
(B) Dexamethasone
(C) Filgrastim
(D) Interferon alfa
(E) Interleukin-2 (IL-2)
(F) Leucovorin

Table 1: An example of an item from the USMLE exam

2008). Different strategies are then employed to select the most plausible distractors and the generated MCQs are most commonly evaluated by experts, and in more rare cases given to students or crowd workers. For example, Papasalouros et al. (2008) present a rule-based approach for selecting the distractors, mostly limiting them to siblings of the correct answer. In another study Žitko et al. (2009) use ontologies to generate the question stems and then propose a random list of alternative answers. More recent approaches make use of the semantics of the domain represented as mapped axioms (Vinu and Kumar, 2015b). Another approach called pattern-based MCQ generation utilizes different combinations of predicates associated with the instances of an ontology to generate the stems (Vinu and Kumar, 2015a). The distractors are selected from the list of instances in the ontology within the intersection classes of the domain or range of the predicates in the stem and are presented in a random order. In a follow-up study, Vinu et al. (2016) manipulate the difficulty of the stem and choice set based on similarity measure called *Instance Similarity Ratio* which takes into consideration the similarity between instances with regards to the conditions in the stem. The system then varies the question difficulty based on the similarity between the distractors, the correct answer and the stem (higher similarity indicates a more difficult question). Evaluation with test-takers revealed a correlation of .79 between the predicted and the actual difficulty levels.

The studies mentioned so far describe automatic and semi-automatic approaches for distractor generation in scenarios where the system generates the entire MCQ (i.e. it controls the stem). In the experiments presented in this paper we introduce a fully automatic approach to distractor generation and selection based on embedding vectors and information retrieval techniques, which can be used for any given stem and correct answer pair. The next section presents the data used in our study.

3 Data

In this study we use multiple-choice questions administered by the United States Medical Licensing Examination (USMLE). The USMLE exam is a high-stakes examination for medical licensure in the United States, the outcome of which is recognised by all medical boards in the USA. The goal of the licensure and certification examination is to ensure that medical professionals have met the required standards and are qualified to engage in practice. The data has been provided by the National Board of Medical Examiners (NBME) who develop and manage the USMLE.

We use two separate data sets of questions where each test item is a single-best-answer multiple-choice question consisting of a stem followed by four or more response options. An example of such item is provided in Table 1.

Our main data set consists of 1,441 multiple choice test items that have been administered or pretested during the 2008 administration of the USMLE. These questions are not available to the public due to test security reasons and are henceforth referred to as the private data set. An additional 369 items which are publicly available[2] have also been used in this study and are referred to as the public data set. The public data set contains 132 questions from the USMLE Step 1 2015 sample booklet, 117 questions from the USMLE Step 1 2016 sample booklet, and 120 questions

[2] The items can be accessed at the USMLE web site, for example: http://www.usmle.org/pdfs/step-1/2017samples_step1.pdf

from the USMLE Step 2 2017 sample booklet. The main characteristics of the test items and their options within both sets are presented in Table 3.

Dataset	Public	Private
Total number of items	369	1441
Total number of options	1728	7664
Total number of distractors	1359	6223
Options per item	4.68	5.32

Table 2: Item characteristics for the two data sets

4 Method

Content specialists are instructed to create distractors that are similar in content and structure relative to the correct options (Ascalon et al., 2007; Gierl et al., 2017; Case and Swanson, 2001). The similarity can be quantified using either ontologies or computational models such as distributional similarity ones. For example, according to embedding vectors which represent the state-of-the-art in distributional similarity, distractors found in actual items are more similar to the correct answers than random concepts; they are also more similar to their stem than a random concept as well (this is also empirically tested further in the paper, see tables 3 and 4). As a result, we extend the instruction that distractors should be similar to the correct answers to computer models used to suggest distractors: distractor candidates are those that are similar to the correct answers and stems, measured using various models of similarity, and specifically, embedding vectors and information retrieval based similarity (Sections 4.1 and 4.3).

We first describe the lexicons, the embedding vectors derived from them (Section 4.1) and how they are used to calculate the similarity between different item parts (e.g. stem, correct answer, answer options, etc.) (Section 4.2). We then describe the methodology for ranking the suggested distractors using information retrieval techniques in Section 4.3.

4.1 The concept embeddings

We use embedding vectors to quantify the similarity between correct answers, distractors, and stems. Precomputed embedding vectors are available for various lexical databases such as Freebase and UMLS. We use the embedding vectors based on data from two lexical-semantic databases:

- Unified Medical Language System (UMLS) [3] 2012. We use the concept embedding vectors provided by Yu et al. (2017). These vectors are built using Pubmed citations published before 2016, bag-of-words model, and 200 dimensions.
- Freebase entities[4]. Freebase is a large collaborative knowledge base containing more than 39 million topics and more than 1.9 billion "facts". We use pretrained vectors for 1.4M entities, trained using 100B words from various news articles[5]. Each vector has 1000 dimensions.

Table 5 shows the number of USMLE item options that are also entries in the two lexical-semantic databases: UMLS and Freebase entities. As can be seen from the table, the UMLS database is a promising source for option candidates, as more than half of the options from both data sets can also be found in this database. On the other hand, Freebase vectors have been derived from much more data compared to UMLS vectors (approximately 100 billion of tokens). Nevertheless, even though Freebase has more concepts than UMLS (the Freebase vectors represent 1.4M entities, whereas UMLS vectors represent 300K concepts), its coverage is poorer in the medicine domain, and only 32% of distractors can be found in the Freebase, versus 56% coverage of UMLS (see Table 5). Based on this comparison, we focus on experimenting with the UMLS vectors and all results reported in the remainder of this paper were obtained using UMLS vectors.

4.2 Similarity calculation

We then calculate the similarity between:

1. The options themselves
2. Distractors and correct answers
3. Stems and options
4. Stems and correct answers

The similarities are calculated using embedding vectors as follows. The embedding vectors map an entity to a vector of n dimensions. In the case of the Freebases entities, n = 1000, and in the case of the UMLS concepts, n = 200. These vectors

[3] https://www.nlm.nih.gov/research/umls/

[4] https://developers.google.com/freebase/

[5] https://code.google.com/archive/p/word2vec/

	Mean	STD	Min	Max	N
Distractor-CorrectAnswer	0.34	0.15	-0.10	0.82	1341
Option-Option (Dist-CorrAns + DistDist)	0.33	0.15	-0.10	0.82	3674
Random pair of entities	0.09	0.13	-0.09	0.92	10000
Stem-Option	0.17	0.08	-0.02	0.53	1860
Stem-CorrectAnswer	0.18	0.08	-0.01	0.53	519
Stem-Random entity	0.05	0.06	-0.13	0.34	1860

Table 3: Cosine similarity between different item-part configurations calculated using Freebase vectors, using the private dataset.

	Mean	STD	Min	Max	N
Distractor-CorrectAnswer	0.41	0.17	-0.12	0.98	2849
Option-Option (Dist-CorrAns + DistDist)	0.39	0.19	-0.21	0.98	7981
Random pair of entities	0.03	0.17	-0.40	0.99	10000
Stem-Option	0.30	0.15	-0.24	0.71	4408
Stem-CorrectAnswer	0.34	0.14	-0.08	0.69	806
Stem-Random entity	0.02	0.17	-0.42	0.64	4408

Table 4: Cosine similarity between different item-part configurations calculated using UMLS vectors, using the private dataset.

	Total hits (%)	
Lexicon	Public	Private
UMLS concepts	964 (56%)	4408 (57%)
Freebase entities	562 (32%)	2734 (36%)
In either	980 (57%)	4448 (58%)

Table 5: Number of USMLE item options that are also entries in the two lexical-semantic databases

represent the distributional information of the entities with regard to some training objective and the cosine distance between two vectors is a good estimation of the similarity between the two entities. Here, "similarity" is defined as the similarity of information the two entities contain that is useful for the objective of the models used to acquire these vectors. The training objectives of the two sets of embedding vectors are to predict the context in which an entity would appear.

The representative embedding of a stem is computed by first translating the stem into a list of Concept Unique Identifiers (CUIs) using Metamap[6]. In cases where numerals were present in the stem (e.g. 100/mm3, 95%), these were excluded. We then sum the CUIs in the stem in the following way[7]:

$$S = L^2 - norm\left(\sum_{CUI in S} V_{CUI}\right)$$

We only choose options that appear in the respective databases. Table 3 shows the cosine similarities calculated using Freebase entities' embedding vectors, whereas Table 4 shows the calculations using UMLS concepts' vectors. We also perform calculations using random entities as a baseline. N represents the number of pairs.

As shown in Table 3, options that are found within the Freebase database are more similar to each other and to the stems, compared to random entities. This suggests that Freebase vectors can be used to suggest option candidates by suggesting entities which are similar to the correct answer. As can be seen from Table 4, options that are found within the UMLS database are also more similar to each other and to the stems than random entities are. The above observations confirm the premise that measurable similarity between distractors and the correct answers as well as the stems can be used as a criterion to suggest distractor candidates. They serve as a basis for our proposed method of predicting which distractor candidates would actually be used, as detailed below.

4.3 Predicting distractors using embedding vector similarity and information retrieval

In order to predict which distractor candidates would actually be used in an item, we first get the list of candidates, and then rank these candidates according to their similarity to the options and stems. The list of candidates could be entire UMLS, or only those that share the same semantic

[6] https://metamap.nlm.nih.gov/

[7] OOV rate: About 8% of the CIUs returned by Metamap do not feature in UMLS embedding

type[8] with the correct answers (STY), are marked as sibling of the correct answer (SIB), or are built using a graph walking method starting from the correct answers, then walk up to their broader concepts and then walk down to the narrower concepts of these broader concepts (RB_RN). For each of these choices there is a trade-off between coverage and precision. Using sibling relation only will produce the least number of candidates, at the expense of having the least coverage (only around 20% of potential matches). On the other hand, using the entire UMLS as candidates would ensure maximum coverage, at the expense of having to consider hundreds of thousands of candidates for each correct answer.

We then sort the candidates according to their similarity to the correct answers combined with the stem. This similarity is measured as the cosine similarity between the embedding vectors of the candidates, and those that represent the sums of the embedding vectors of the correct answer and those of the stem.

Top 10, 20, and 100 are called "predictions", and the number of correct "predictions" (i.e. the number of candidates that actually features as real distractors) is recorded as hits.

We also incorporate information retrieval. We first get the top n suggestions (in our experiment, we use $n = 500$), as previously described, we then rerank the candidates according to the rank of the first document in which they appear, when we use the stems as the query as we search our text collection, in our case, 2013 MEDLINE citations[9]. We use Lucene[10] for indexing and retrieving documents. The premise for this reranking is similar to that of Mitkov and Ha (2003): distractor candidates that appear in the same document that contains fragments of the stem would be prioritised over other candidates. Documents that contain fragments of the stem are retrieved by querying the text collection with the stem as the query.

To the best of our knowledge, a similar set up for the generation and evaluation of distractors has not been proposed before, which is why we are not able to compare our results to baselines from previous studies. We do, however, compare the performance of our system to a baseline of random hit prediction. Furthermore, the concept-embedding approach can be viewed as a baseline compared to the approach using concept embeddings combined with IR techniques.

5 Evaluation

In order to evaluate our approach and the usefulness of the suggestions in the generated list, we describe an evaluation procedure where our system takes existing items together with all their options and tries to "predict" one or more of the existing distractors. In other words, if the system comes up with one or more of the same distractors as the ones produced by the human item-writers, then the approach could be considered useful for the generation of suitable distractor suggestions for new items. To do this, for each item, we get the first *n* concepts that are most similar to the combination of stem and the correct answer, and see how many of these concepts actually feature as distractors in that item (hits). The number of hits provides an estimation of the usefulness of the suggested list.

The results are presented in Table 6. Within that table, *Applicable items* are the ones whose correct answers could produce distractor candidates using the specific ontology relation. *Number of all candidates* reflects the number of candidates suggested by the specific ontology relation. *Maximum number of hits* refers to the number of hits if all the suggested candidates are considered, *Random N hits* is the number of hits if random N candidates are picked for each item. *Recall at N* signifies the total number of hits if the top N candidates are considered, divided by the total number of distractors that also feature in UMLS. In terms of ontology relations, SIB includes only candidates that are considered to be the siblings of the correct answer (according to UMLS). RN_RB means that only candidates that share a broader or narrower concept with the correct answer are considered, and STY means that all candidates that share the same semantic type with the correct answer are considered. The precision and recall relation is presented in Figure 1, while Figures 2 and 3 present the recall for the private and public data sets respectively.

As can be seen from Table 6, the suggested list outperformed the baseline of random hits in all three types of relations (SIB, RB_RN and STY), where best result (in terms of trade off between precision and recall) is achieved for the top 20 hits. Using the broadest ontology relation, namely

[8] For the full list of semantic types in UMLS, please visit https://www.nlm.nih.gov/pubs/factsheets/umlssemn.html

[9] https://mbr.nlm.nih.gov/Download/Baselines/2013/

[10] https://lucene.apache.org/

By approach (the relation used is STY)

		Public	Private
Top 10	Embedding only	73	319
	IR reranking	76	325
	Improvement	4%	2%
Top 20	Embedding only	99	492
	IR reranking	142	572
	Improvement	43%	16%
Top 100	Embedding only	190	811
	IR reranking	275	1242
	Improvement	45%	53%

By Ontology relation (IR reranking is used)

	Public			Private		
Ontology relations	SIB	RB_RN	STY	SIB	RB_RN	STY
Applicable items	143	165	181	640	756	806
Number of all candidates	3657	63998	10804667	18660	327623	48539316
Maximum number of hits	85	208	473	424	942	2360
Top 10 hits baseline	75	70	76	333	290	325
Random 10 hits	57	11	1	275	64	2
Recall at 10 (over all possible UMLS distractors, see last row)	0.13	0.12	0.13	0.12	0.10	0.11
Top 20	82	120	142	382	450	572
Recall at 20	0.14	0.20	0.24	0.13	0.16	0.20
Top 50	85	165	233	410	750	968
Recall at 50	0.14	0.28	0.39	0.14	0.26	0.34
Top 100	85	191	275	415	844	1242
Recall at 100	0.14	0.32	0.46	0.15	0.30	0.44
Distractors that belong to items whose correct answers feature in UMLS, and themselves also feature in UMLS	592			2831		

Table 6: Evaluation results: distractor hits.

same semantic type (STY), performs as well as the *sibling* (SIB) relation for the top 10 hits (i.e. 76 vs. 75, respectively, for the public data set and 325 vs. 333 for the private one). From the top 20 hits onwards, the STY relation outperforms SIB (i.e. for 20 hits we have STY hits= 142 and SIB = 82 for the public data set and STY = 572 and SIB = 382 for the private data set).

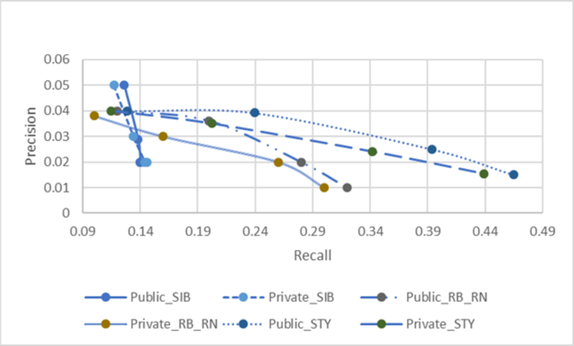

Figure 1: Precision - Recall Relation Graph

An example of a question and the list of generated distractors and their ranking is presented in Table 7. As can be seen from the table, the

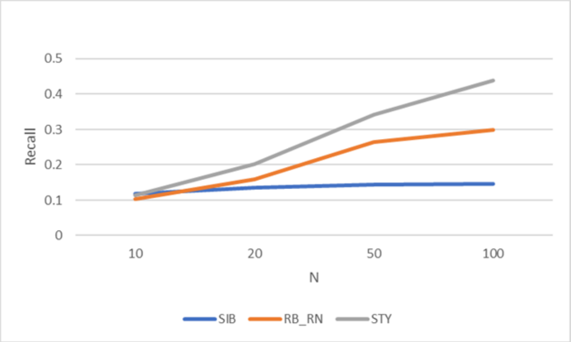

Figure 2: Recall at N, Private Data Set

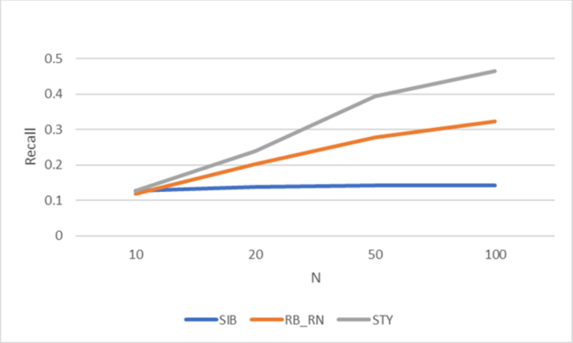

Figure 3: Recall at N, Public Data Set

Example Question 85, 2015 Booklet
A 30-year-old man with peptic ulcer disease suddenly develops pain, redness, and swelling of his right first metatarsophalangeal joint. There is no history of injury. Serum uric acid concentration is 8 mg/dL. Examination of joint aspirate shows birefringent crystals. Which of the following drugs is most appropriate to treat the acute symptoms in this patient? (A) Allopurinol; (B) Colchicine (correct answer) (C) Morphine; (D) Probenecid; (E) Sulfinpyrazone

SIB Top 10	SIB Top 10 IR	RB_RN top 10	RB_RN top 10 IR	STY top 10	STY top 10 IR
Vinca alkaloid	Allopurinol*	Desacetylcolch.	Allopurinol*	Colchicoside	Probenecid*
Castanospermine	Probenecid*	Colchamine	Morphine*	Cornigerine	Indomethacin
Emetine	Opioid	Vinca alkaloid	Probenecid*	Vinblastine sulf.	Benemid
Probenecid*	Quinine	Thiocolchicoside	Indomethacin	Desacetylcolch.	Sulfinpyrazone*
Cyproheptadine	Dl-hyoscyam.	Desmethylphall.	Naproxen	Oncodazole	Gabexate Mesylate
Strychnine	Amitriptyline	O-methylandroc.	Sulfinpyrazone*	Lumicolchicine	Deltahydrocort.
Swainsonine	Cocaine	Isocolchicine	Uricosuric agent	VLB	Cholestyramine res.
Staurosporine	Emetine	Chelidonine	Methyl morphine	Oryzalin	Methotrexate
Paclitaxel	Hyoscine	Tropone	Opioid	Demecolcine	6-alpha-Methylp.
Aconitine	Nicotine	Paclitaxel	Quinine	Colchicine analog	Ursodeoxycholic Ac.

Table 7: Example of the Top 10 candidates suggested by various ontological relations and rankings for Question 85 from the 2015 booklet. Suggestions that also feature in the item are marked with *.

information-retrieval ranking improves the number of hits in all types of relations (SIB, RB_RN, and STY). It should be noted that the improvement we notice in this example is not as significant in other examples but the general trend is the same. The average improvement across of all items can be seen in Table 6.

6 Discussion

The results presented above indicate that best performance is achieved when combining the two approaches, namely generating distractors using concept embedding similarity to provide the initial list, and then using a re-ranking approach from information retrieval in order to improve the prediction. Using this combined approach, our system can hit around 1 in 5 distractors produced by the human-item writers when producing 20 candidates for each item. It should be noted that a random pick in the case of "predicting" distractors has a very low chance of being correct. For example, using the STY relation, a random 10 chosen distractor candidates for each item will probably produce ***one*** hit for the whole public dataset, and ***two*** hits for the whole private dataset. It is also worth noting that the proposed method does not rely on training data.

It was shown that the *STY* relation outperformed the *SIB* relation in the samples of top 20, top 50, top 100 hits. The reason for this result is the ability of the *STY* relation to consider more candidates. Based on these results, we recommend the use of a broader ontology relation. Further to this, the results presented in Table 6 indicate that the longer the list of suggested ditractors, the smaller the return. As can be seen, the return diminishes when having a list of more than 20 suggested distractors.

One limitation of the current evaluation is the fact that it assumes that the distractors developed by the human item-writers are the best ones. As shown in the introduction section, this may not necessarily be the case since item-writers also find the selection of plausible distractors a challenging task. It is also quite possible that some of the automatically generated distractors are suitable enough even though they were not included as an item option and in this sense it is possible that our evaluation has been too conservative and that more distractor candidates are in fact feasible options. To address this we plan a future evaluation where human item-writers will be presented with a list of automatically generated distractors that they can choose from. An even longer term evaluation would be to assess the quality of the distractors by collecting data from examinees and using the item response theory (Embretson and Reise, 2013). Another limitation is that since we do not have control over the stem, we do not control for cases where a plausible distractor candidate may

in fact be an alternative correct answer. To a certain extent this is mitigated by the condition that no synonyms of the correct answer can feature as distractors and that, ultimately, there would be a human item-writer who selects the most suitable distractors proposed by the system.
To the best of our knowledge, the experiments presented in this paper are the first fully automatic approach for distractor generation which relies on the combination between concept embeddings and IR. The benefit of this approach is not only its performance but that it can also be generalized to other domains where the distractors are concepts.

Directions for improvement include experimenting with different embedding vectors or ontological relations (such as RO (other relation) in UMLS). In addition, instead of using the whole stem as the query to search the text collection, one could break the stem into smaller components, and search using these components[11] Last but not least, the number of prediction hits could be enhanced through other machine learning models.

7 Conclusion

We presented an experiment for the automatic suggestions of distractors for multiple-choice questions given a question stem and the correct answer option. Our method was based on concept embeddings and re-ranking of the distractors candidates using an information retrieval approach. To evaluate the output, we compare the existing human-generated distractors and the automatic suggestions in two sets of items. The results indicate that the concept embeddings can correctly predict one in five possible distractors, which otherwise has a very low chance of being predicted randomly. Reranking of the candidates boosts the performance significantly, which shows that approaches from IR can contribute to the task of automatic distractor generation.

References

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

Tahani Alsubait, Bijan Parsia, and Ulrike Sattler. 2013. A similarity-based theory of controlling mcq difficulty. In *e-Learning and e-Technologies in Education (ICEEE), 2013 Second International Conference on*, pages 283–288. IEEE.

M Evelina Ascalon, Lawrence S Meyers, Bruce W Davis, and Niels Smits. 2007. Distractor similarity and item-stem structure: Effects on item difficulty. *Applied Measurement in Education*, 20(2):153–170.

Krista Breithaupt, Adelaide A Ariel, and Donovan R Hare. 2009. Assembling an inventory of multistage adaptive testing systems. In *Elements of adaptive testing*, pages 247–266. Springer.

Jonathan C Brown, Gwen A Frishkoff, and Maxine Eskenazi. 2005. Automatic question generation for vocabulary assessment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 819–826. Association for Computational Linguistics.

Susan M Case and David B Swanson. 2001. *Constructing written test questions for the basic and clinical sciences*. 3rd edition. National Board of Medical Examiners Philadelphia.

Susan E Embretson and Steven P Reise. 2013. *Item response theory*. Psychology Press.

Mark J Gierl, Okan Bulut, Qi Guo, and Xinxin Zhang. 2017. Developing, analyzing, and using distractors for multiple-choice tests in education: a comprehensive review. *Review of Educational Research*, 87(6):1082–1116.

Mark J Gierl, Hollis Lai, James B Hogan, and Donna Matovinovic. 2015. A method for generating educational test items that are aligned to the common core state standards. *Journal of Applied Testing Technology*, 16(1):1–18.

Mark J Gierl, Hollis Lai, Debra Pugh, Claire Touchie, André-Philippe Boulais, and André De Champlain. 2016. Evaluating the psychometric characteristics of generated multiple-choice test items. *Applied Measurement in Education*, 29(3):196–210.

Michael Heilman. 2011. *Automatic factual question generation from text*. Ph.D. thesis, Carnegie Mellon University.

Edmond Holohan, Mark Melia, Declan McMullen, and Claus Pahl. 2006. The generation of e-learning exercise problems from subject ontologies. In *Advanced Learning Technologies, 2006. Sixth International Conference on*, pages 967–969. IEEE.

Ayako Hoshino and Hiroshi Nakagawa. 2005. A real-time multiple-choice question generation for language testing: a preliminary study. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 17–20. Association for Computational Linguistics.

[11] In the case of our sets of items, the components could be “chief complaint, further history, significant positives, significant negatives, medical history, medications, physical examination, lab values”.

Hollis Lai, Mark J Gierl, Claire Touchie, Debra Pugh, André-Philippe Boulais, and André De Champlain. 2016. Using automatic item generation to improve the quality of mcq distractors. *Teaching and learning in medicine*, 28(2):166–173.

Mukta Majumder and Sujan Kumar Saha. 2015. A system for generating multiple choice questions: With a novel approach for sentence selection. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pages 64–72.

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.

Andreas Papasalouros, Konstantinos Kanaris, and Konstantinos Kotis. 2008. Automatic generation of multiple choice questions from domain ontologies. In *e-Learning*, pages 427–434. Citeseer.

Marie Tarrant, James Ware, and Ahmed M Mohammed. 2009. An assessment of functioning and non-functioning distractors in multiple-choice questions: a descriptive analysis. *BMC medical education*, 9(1):40.

Ellampallil Venugopal Vinu, Tahani Alsubait, and P Sreenivasa Kumar. 2016. Modeling of item-difficulty for ontology-based mcqs. *arXiv preprint arXiv:1607.00869*.

Ellampallil Venugopal Vinu and P Sreenivasa Kumar. 2015a. Improving large-scale assessment tests by ontology based approach. In *FLAIRS Conference*, page 457.

EV Vinu and Sreenivasa Kumar. 2015b. A novel approach to generate mcqs from domain ontology: Considering dl semantics and open-world assumption. *Web Semantics: Science, Services and Agents on the World Wide Web*, 34:40–54.

Zhiguo Yu, Byron Wallace, Todd Johnson, and Trevor Cohen. 2017. Retrofitting concept vector representations of medical concepts to improve estimates of semantic similarity and relatedness. In *Proceedings of MedInfo - World Congress on Medical and Health Informatics*, pages 657–661. International Medical Informatics Association.

Branko Žitko, Slavomir Stankov, Marko Rosić, and Ani Grubišić. 2009. Dynamic test generation over ontology-based knowledge representation in authoring shell. *Expert Systems with Applications*, 36(4):8185–8196.