CS 510 Advanced Information Retrieval

MP 2

Problem 1 (c)

(i)

Given:

e_x be the unit-length embedding for X

e_y be the unit-length embedding for Y

e_z be the unit-length embedding for Z

We can construct the scoring function for ranking words to complete the analogy 'X is to Y as Z is to __' as follows:

We compute the query vector $q = e_y - e_x + e_z$ and find the most similar word vector $v \in E$ (Embedding matrix) to q.

To do so, we normalize q to have unit length i.e. $q_1 = \frac{q}{||q||}$ and assuming each vector in embedding matrix E is also unit length, it reduces to computing:

$$\text{argmax}_{v \in E}\ v \cdot q_1$$

and returning the associated word v.

Hence, we compute $q = e_y - e_x + e_z$ and return vector v with $\text{argmax}_{v \in E} = v \cdot q_1$ where $q_1 = \frac{q}{||q||}$.

(ii)

Upon querying 'man woman king', I expected that the ranking results would be such that some of the words from the query and the actual result 'queen' would be on top and after deleting the query words, 'queen' would be among the top 3 at least.

For SVD with p=1, I got 'queen' at the last rank (10^{th}) with score = 0.8384, which is not a good result since the desired word came last.

For SVD with p=0.5, I got 'queen' at the fifth rank (5^{th}) with score = 0.74330, which is not a very satisfactory result since the desired word came fourth, even after deleting the query word 'king' from the ranked list.

For SVD with p=0, I got 'queen' at the fourth rank (4^{th}) with score = 0.691775, which is kind of okay result since the desired word came third after deleting the query word 'king' from the ranked list.

Note that I got the query word 'king' ranking first in all the three methods.

So, from this one example, we can observe that SVD with p=0 is performing best out of three and SVD with p =1 performed worst of all. Also, that this method is not very accurate and is not able to capture the word relationships well. This means that the vector space representing these words and their relationships is not yet defined accurately and completely. Some relationships are still not captured perfectly. It seems that the method has shortcomings and can be improved.

Next, I tried to query the following different analogies for the SVD methods with p=0, 0.5 and 1 and the results **(ranks are after deleting the query words from the ranked list)** are tabulated below:

Analogy tested	Desired word	SVD with p=1 (rank, score)	SVD with p=0.5	SVD with p=0
Woman is to sister then man is to	Brother	6^{th}(0.91368)	4^{th} (0.8524163)	4^{th} (0.8039287)
Summer is to rain then winter is to	Snow	1^{st} (0.9090)	1^{st} (0.8778493)	1^{st}(0.85180)
Fell is to falling then ate is to	Eating	1^{st} (0.90678)	1^{st} (0.8426187)	1^{st} (0.787734)
Running is to ran then crying is to	Cried	3^{rd}(0.81544)	3^{rd} (0.727569)	3^{rd} (0.685287)
Man is to husband then woman is to	Wife	3^{rd} (0.938813)	4^{th} (0.906568)	4^{th} (0.8895441)

Notice that it can observed from the small sample set of analogies that we have tested to judge the performance of the SVD methods with different values of p that the method does not perform very well with relationship queries. It can be clearly noticed that qualitatively, the method can be improvised to produce better results. It is not able to currently capture the word dependencies perfectly. But none the less, out of all the queries that I tried, the desired word was in top 6 at least all the time, hence we can say that the method does work but can be improved more to analyze and capture word dependencies perfectly. This actually means that the vector space representing the words and their relationships is not yet defined accurately and completely. Some relationships are still not defined accurately and hence we

are not able to capture them. Though for some other relations, the method works fine because the part of vector space corresponding to them has been defined well.

Qualitatively, it performs well on some queries whose vector space is defined well and bad on other whose vector space is still to be constructed completely.

Problem 2

(a)

Mean reciprocal rank (MRR) is average of the reciprocal of the rank of the top desired result over a set of topics.
Accuracy is the total no. of queries that resulted in desired result at top of the ranked list after removing query words from the ranked list over total no. of queries run.

The preference of accuracy or mean reciprocal rank depends on users' tasks and preferences.

Consider the case where there is precisely one relevant document in the whole collection. As an example, consider known item search on Amazon or Facebook or question answering where there is only one answer. In these scenarios, the goal is to retrieve the top ranked answer.
If we rank the answers, then the goal is to rank that one particular answer on the top. In this case, the average precision will reduce to the reciprocal rank $\frac{1}{r}$ where r is the rank of the single relevant document.

If that document is ranked on the very top, then the reciprocal rank would be 1. If it's ranked at the second position, then it's $\frac{1}{2}$ etc. Hence, mean reciprocal rank (MRR) i.e. average of all the reciprocal ranks over a set of topics is a useful metric in these cases.

Intuitively, r indicates how much effort a user would have to make in order to find that one relevant document.
Directly using r to measure the performance of a system where there is only one relevant item is not feasible since if we average over a large number of topics, sum of r would be dominated by large values of r. Note that large values or r basically indicate lower ranked results i.e. the average would then be dominated by the relevant documents that are ranked lower in the list.
But clearly, users care more about the highly ranked documents, so by taking reciprocal rank we emphasize more on the top ranked documents. Difference between say rank 100 and 1000 and rank 1 and 2 using each method intuitively clarifies the use of MRR in the

scenarios of single relevant document while taking average over large no. of topics. Since, taking rank 1 and 2 would result in a big difference in rank $\frac{1}{r}$ but not in r. Taking rank 100 and 1000 would result in big difference in r, but not in rank $\frac{1}{r}$. And since user cares more about highly ranked documents than others, rank $\frac{1}{r}$ is an appropriate measure. Hence, we prefer Mean Reciprocal Rank (MRR) over accuracy.

Hence, MRR is preferred and is more informative than accuracy if there is only one relevant item in the collection example, known item search or question answering with only one relevant answer.

Next, consider accuracy:
MRR only cares about the position of the first relevant document so if we care about many relevant documents as possible to be ranked high on the ranking list, MRR should not be the choice. Therefore, MRR is appropriate to judge a system where either there's only one relevant result or user really cares about the one highest ranked.
Accuracy, on the other hand considers all of the relevant items that tend to get ranked first. If more than one document is considered the most relevant, and the query reveals any one of them as the top one, it is considered a correct solution. Then accuracy is the total no. of queries that resulted in desired result at top of the ranked list after removing query words from the ranked list over total no. of queries run. Hence, accuracy is used over MRR when there are multiple most relevant solutions.

For example, there's no need to use accuracy if we have only 1 relevant answer, MRR works fine. But if we have a query such as "Presidents of US" and the top three results are "Donald Trump", "Barack Obama", and "George W. Bush, distinct but all are correct answers, we cannot use MRR because there are multiple most relevant solutions and there is no measure to decide whose rank 'r' we should use in the ranked list to compute the reciprocal rank. But, we can measure the accuracy by running query and if any of the three most desired answers come at top, we consider it as a correct solution, else incorrect.

Hence, accuracy is preferred over MRR if there are multiple most relevant items in the collection for the query. Since if any one of them comes at top, we assume a correct solution for the query while computing accuracy.

Also, accuracy captures the ratio of relevant and non-relevant documents that were correctly classified as relevant and non-relevant respectively (True positive + True negatives) to all the relevant and non-relevant documents (True positive + True negative + False positive + False negative).
So, we can also use accuracy as a metric to measure correctly classified answers to all the answers. This is again where MRR cannot be used since MRR only captures the rank of the most desired answer and uses the same to judge the accuracy of the method, but on the

other hand accuracy uses the classification of all the answers and uses that to judge the correctness of the method.
Hence, accuracy is preferred over MRR when multiple relevant documents are present in the collection corresponding to the query and when we want to judge the quality of the method based on all the classifications of relevant and non-relevant answers it did rather than just looking at the rank of the most desired answer.
So, if we have a query such as "Presidents of US" and the top three results are "Donald Trump", "Barack Obama", and "George W. Bush, distinct but all are correct answers, we can clearly quantify accuracy as the more desired method since it captures the information about all three most desired answers coming at top, compared to MRR that just floated one out of these most desired answers as the top most.

Hence, accuracy is preferred over MRR if there are multiple most relevant items in the collection for the query. Since accuracy can capture the information about all the desired relevant answers in comparison to MRR that just float one of the desired answers as the top most.

(c)

The MRR for the different embeddings method is as follows:

Method	SVD (p=1)	SVD (p=0.5)	SVD (p=0)	GloVe
MRR	0.360165 (+/- 0.41878)	0.396301 (+/- 0.42689)	0.39561 (+/- 0.42637)	0.49357 (+/- 0.4319)

Ignoring the standard deviation term, we have:

Method	SVD (p=1)	SVD (p=0.5)	SVD (p=0)	GloVe
MRR	0.360165	0.396301	0.39561	0.49357

(i)

We notice that SVD of the embedding matrix E = $U_k(\sum_k)^p$ with p=0.5 performs the best (MRR = 0.396) of all SVD with values p=1 (MRR = 0.360), p=0.5 and p=0 (MRR = 0.395).

The performance of SVD deteriorates significantly with p=1. By using the traditional way of computing the embeddings using SVD by letting the embedding matrix E = $U_k\sum_k$ performs the worst. Despite being theoretically motivated, this setting with p=1 has very poor results in practice when compared to p=0.5 or p=0.

The above table demonstrates the MRR results for each of the settings. The drop in average accuracy by setting p=1 is surprising. This is the reason we must choose p = 0.5 as the default setting for the SVD in the vanilla scenario i.e. the embeddings must be computed using SVD by letting the embedding matrix E = $U_k(\sum_k)^{0.5}$.

(iii)

Notice that from part (i), the two best performing methods in terms of MRR are GloVe (MRR = 0.49) and SVD with p = 0.5 (MRR=0.396).

We see that the same individual (reciprocal rank for each query) is measured twice for two different methods and we need to check if the mean of the reciprocal ranks for GloVe method is statistically significantly better from the mean of the reciprocal ranks of SVD with p = 0.5 method.

This is a hypothesis testing problem using one sided dependent t – test. The hypothesis test is a paired-samples t-test because we have two ranked lists, and all queries are in both the ranked lists. We need to check if the MRR of the best model we observed i.e. GloVe is statistically significantly better than the MRR of the second-best model we observed i.e. SVD with p = 0.5.

Let Reciprocal $Ranks_1$ be the reciprocal ranks for best method i.e. GloVe
Reciprocal $Ranks_2$ be the reciprocal ranks for second best method i.e. SVD with p = 0.5

Consider the null hypotheses,
H_0: E (Reciprocal $Ranks_1$) = E (Reciprocal $Ranks_2$)

And the alternate hypothesis,

H_1: E (Reciprocal $Ranks_1$) > E (Reciprocal $Ranks_2$)

According to the results of the two sided t – test in analogies.py for GloVe method and SVD with p = 0.5 method, we get that t statistic for the test is 18.545821 and the p value for the test is $9.26021*10^{-75}$.

Since, t value > 0, and we want to test for **one sided alternate hypothesis E (Reciprocal $Ranks_1$) > E (Reciprocal $Ranks_2$)**, test statistic value remains the same but the p value for the test is the half of the computed p value using 2 tailed test.

Hence, p value for the one tailed t-test is $4.6301*10^{-75}$

- **t statistic for the test = 18.545821**
- **p value for the test = $4.6301*10^{-75}$**

Now, notice that p value is extremely small, especially compared to significance levels

$\alpha = 0.1 \; or \; 0.05$ (i.e. 90% or 95% confidence interval). Hence, we will reject the null hypothesis for significance levels $\alpha = 0.1 \; or \; 0.05$.

Hence,
Yes, according to the t-test, we conclude that the MRR values of GloVe are statistically significantly better from the mean of the reciprocal ranks of SVD with p = 0.5 method.

Problem 3

(b)

The win ratio for the different embedding methods using user evaluation are:

Method	SVD (p=1)	SVD (p=0.5)	SVD (p=0)	GloVe
Win Ratio	0.534482758621	0.568965517241	0.51724137931	0.431034482759

The highest win ratio is from the method SVD with p=0.5.

Hence, we conclude that on using user task for evaluating word embedding methods, method SVD with p=0.5 performs the best.

No, this is not what I expected. According to the previous results in Problem (2) and knowledge, I expected GloVe to perform the best and SVD with p=1 to perform the worst. But the results of the user task for evaluating embedding methods are not agreeing. We see that the win ratio for GloVe is least and for SVD with p=0.5 is highest. If we use a large group of users, the average is likely to perform better than just taking the results from one user and will be less user biased. The major reason for this decrepency results can be user biasedness because the results are just based on my own (only 1 user's) judgements.

(c)

Notice that from part (b), the two best performing methods in terms of evaluating word embedding methods using user task are SVD with p=0.5 (Win ratio = 0.568965517241) and SVD with p=1(Win ratio = 0.534482758621).

We see that the same individual (win ratio for each query) is measured twice for two different methods and we need to check if the win ratios for SVD with p=0.5 method is statistically significantly better from the win ratio of SVD with p=1 method.

This is a hypothesis testing problem using one sided dependent t – test. The hypothesis test is a paired-samples t-test because we have two win lists, and all queries are in both the win lists. We need to check if the win ratio of the best model we observed i.e. SVD with p=0.5 method is statistically significantly better than the win ratio of the second best model we observed i.e SVD with p=1.

Let Win_1 be the wins (1 for win, 0 for loss) for best method i.e. SVD with p=0.5

Win_2 be the wins (1 for win, 0 for loss) for second best method i.e. SVD with p=1

Then for the null hypotheses, consider the win ratio for the best and second-best methods:

H_0: E (Win_1) = E (Win_2)

And the alternate hypothesis,

H_1: E (Win_1) > E (Win_2)

According to the results of the two sided t – test in similarity.py for SVD with p=0.5 method and SVD with p=1 method, we get that t statistic for the test is 0.531204497 and the p value for the test is 0.5973410765.

Since, t value > 0, and we want to test for one sided alternate hypothesis

E (Win_1) > E (Win_2), test statistic value remains the same but the p value for the test is the half of the computed p value using 2 tailed test.

Hence, the p value for the one tailed t-test is 0.29867050.

- **t statistic for the test = 0.531204497**
- **p value for the test = 0.29867050**

Now, notice that p value is large, especially compared to significance levels $\alpha = 0.1 \; or \; 0.05$ (i.e. 90% or 95% confidence interval). Hence, we will cannot reject the null hypothesis for significance levels $\alpha = 0.1 \; or \; 0.05$.

Hence,
No, according to the t-test, we conclude that the Win ratio values of SVD with p=0.5 are not statistically significantly better from the win ratio values of SVD with p = 1 method.

(d)

Clearly while evaluating, I was confused sometimes because I felt two or more words were equally relevant.

Since we are using binary choices, this type of user study setup has a potential problem when a user thinks two or more words in the list are equally good and user is allowed to select only one out of them. This may be because of a variety of reasons, namely:

1. Due to the lack of the domain knowledge of the user about the particular word
2. Because the words ranked at top by different methods may be from different contexts related to the query word and each of the context makes sense, but user is unable to decide what context he/she should judge the results on
3. Because for user, these two or more options were equally or almost equally relevant to the query word, even in the same context, but he had to randomly pick one, because the model allows to choose exactly one option.

For example:

1. For the word upset, according to the four models, two highly ranked words were disappointed and embarrassed. Both words are equally relevant to upset according to me, but since I had to choose only 1, I randomly picked 'disappointed'.
2. For the word armstrong, according to the four models, three highly ranked words were roth, johnston and stevens. Due to lack of domain knowledge, I was unable to decide which one is most relevant word and I felt all are but again I made a random choice to pick roth.

Some of the modifications that may help improve the user study design are as follows:

1. Because this user study was based solely on one user, the results are biased. But if we use a large group of users, even if some of the users will like two or more options and select one randomly, due to informed decision of other users, the average is still likely to prefer the most relevant words.
2. Since in this model, we can choose only 1 even if we feel some are equally relevant, we have to ultimately randomly choose one. So, we may allow user to select more

than one option in case he/she feels more than one are equally relevant hence incrementing the score (and the total average) for all the methods used to produce the relevant options for a particular query word.

3. Instead of using binary choices, we can implement ratings or weights so if a user feels two or more options are relevant, rather than giving a binary choice of selecting 1, he can rate the relevant ones based on a fixed scale say 1-10. This will increment the final score for all the relevant options user felt, and ratings help to even more precisely judge the relevance of words to the query word.

4. If along with the similarity words, we provide the context in which they are similar to the query word as well, it will make it easier for the user to break the ties between words he thinks are equally relevant. For example, for some of the query words, two most relevant words were say the query word's synonym and antonym, then given this relationship or context with the query word, it will be easier for the user to select the context he feels is more relevant to him, either similar or opposite word meaning in this example.

Hence, adding more users to the study, using ratings on a fixed scale rather than binary choices and allowing the user to rate equally two or more words he felt are equally relevant (repeating his ratings) can improve the model in terms of judging the relevance of the words to the query word. Also providing the context or relationship of the choice word with the query word along with it can help user make his decision when he feels two or more choices are equally relevant.

(e)

While evaluating, I sometimes felt that none of the words were relevant to the query.

Since we are using binary choices, this type of user study setup has a potential problem when a user thinks none of the words in the list are good enough and user is still forced to select one out of them. This may be because of a variety of reasons, namely:

1. Due to the lack of the domain knowledge of the user about the particular word, user feels none of the words is of relevance.
2. Because the words ranked at top by different methods may be from different contexts related to the query word but user feels that none of these contexts is of relevance.

For example:

1. For the word shame, according to the four models, two highly ranked words were ! and hate. Both words are not relevant to shame according to me, but since I had to choose only 1, I randomly picked 'hate'.

Some of the modifications that may help improve the user study design are as follows:

1. Because this user study was based solely on one user, the results are biased. But if we use a large group of users, even if some of the users will feel none of the options are relevant and randomly pick one, due to informed decision of other users, the average is still likely to prefer the most relevant words.

2. Since in this model, we must choose 1 option, even if we feel all are irrelevant, we have to ultimately randomly choose one. So, we may allow user to select no option in case he/she feels none of them are relevant hence decrementing the score (and the total average) for all methods used to produce the irrelevant word for a particular query word. This makes sense, because if all the methods produced an irrelevant word, hence, the score for each of them should decrement, but since this query word didn't reveal which method works better over others, ranking doesn't change (but score of all decrements).
3. Instead of using binary choices, we can implement ratings or weights so if a user feels none of the options are relevant, rather than giving a binary choice of selecting 1 random option, he can rate the irrelevant ones very low based on a fixed scale say 1-10. This will decrement the final score for all the irrelevant options user felt, and ratings help to even more precisely judge the irrelevance of words to the query word.

   Hence, adding more users to the study, using ratings on a fixed scale rather than binary choices and allowing the user to rate equally very low to all the options present in case he/she feels none of the words were relevant to the query word can improve the model in terms of judging the relevance of the words for the query word.

Submitted by:

Rachneet Kaur, NET ID: rk4