Capturing Semantic Relevance with DSSM

TAL Future Camp Project Write-up

Qing Lyu
Tsinghua University
lvq15@mails.tsinghua.edu.cn

Abstract

This is my project write-up in TAL Future Camp 2018[1], an AI boot camp organized by a Education and Technology Enterprise in China, Tommorrow Advancing Life Education Group (NYSE: TAL) [2]. The camp provided an intense 2-week project-based training for aspiring AI + Education practitioners, and the division I joined was NLP. Specifically, this project focuses on a semantic relatedness capturing task under an educational setting. Given a pair of elementary school maths exercise problems, it aims at building a system capable of automatically determining if they are conceptually related. As a baseline, I re-implemented an adapted version of the Deep Structured Semantic Models (DSSM) proposed by Microsoft Research in 2013. On top of that, I experimented with several extensions (from the perspective of pre-processing, regularization, architecture design, and parameter tuning) in order to refine the baseline model. In addition to quantitative evaluation, a text retrieval task is carried out to demonstrate the model performance qualitatively. A section of further thoughts is presented in the end, showing current limitations and future possibilities of improvement. All data and codes can be found here[3].

1 Introduction

1.1 Background

The technique of semantic relatedness computation has multiple applications, such as search engines (given a query, retrieving relevant Web documents), question and answering systems (selecting pertinent answers for a certain question) and so on. In order to judge semantic relevance, a prerequisite is to learn good semantic representations for sentences. Distributional semantics provides traditional approaches such as Bag of Words, Word2Vec, and Doc2Vec. In addition,

[1] https://future.ailab.100tal.com/camp/main
[2] http://en.100tal.com
[3] Project repository: https://github.com/veronica320/Capturing-Semantic-Relevance-with-DSSM

deep learning can be another way to learn sentence representations and perform semantic relatedness calculation. In the educational context, we will specifically focus on the task of determining the relevance of a pair of exercise problems, which have potential downstream applications such as exercise recommendation in adaptive learning. We will try to adapt a classical system, Deep Structured Semantic Models (DSSM) (Huang et. al., 2013[1]), and its variation, Convolutional Deep Structured Semantic Models (CDSSM) (Shen et. al., 2014a[2], Shen et. al., 2014b[3]), proposed by Microsoft Research, and apply them under our task settings.

1.2 Task Description

Given a corpus of elementary school maths problems (provided by TAL), our goal is to determine if there exists relevance between any two problems. Instead of other subjects, only maths is chosen, because it is believed to be more complex in text form (with Chinese characters, English and Greek letters, LaTeX notations, equations, and so on), and thus more challenging as well as interesting.
In the dataset, each problem is paired with multiple "related" problems (annotated as "1") and the same number of "unrelated" problems (annotated as "0") at the same time. In detail, our tasks are as follows:

- Build a deep semantic model with the training data, and learn vector representations for the problem texts.
- Use the learned model to predict the relatedness of each pair in the test set.
- Based on the learned semantic representations, draw a few sample problems, and retrieve the "10 most related problems" with each of them according to vector distance.

2 Methodology

2.1 Dataset

The dataset is a Chinese corpus of elementary school maths exercise problems, provided by TAL. As mentioned above, each problem is paired with an equal number of conceptually related problems and unrelated problems. There are 20286 pairs in total, 88.82% for training, and 11.18% for testing. The problems are all written in uncleaned texts, i.e., they contain Chinese characters, Arabic numerals, English and Greek letters, LaTeX notations, equations, and so on. A sample text is:

```
两箱 水果 共重 $ $ 92 $ $ 千克 ， 第一 箱比 第二 箱重 $ $ 6 $ $ 千克 ， 两箱 水果 各重
多少 千克 ？ $ $ left （ 92 + 6 right ） div 2 = 49 $ $ （ 千克 ） $ $ cdotscdots
$ $ 第一 箱 ， $ $ 92 - 49 = 43 $ $ （ 千克 ） $ $ cdotscdots $ $ 第二 箱 .
```

Figure 1: A sample problem text from the dataset

The text is already tokenized. Conspicuously, it needs further pre-processing before being fed to models. The pre-processing strategies will be introduced in Section 2.1 and Section 3.2.1.

2.2 Pre-processing

Dimensionality thresholding. In the corpus, since texts contain different number of words, there will be a problem of unequal vector lengths if directly fed into neural networks. In the papers on DSSM and CDSSM by Microsoft Research, the authors address the problem by either word hashing, or bag of words plus max pooling. In Chinese, however, there is no "curse of dimensionality" , and thus the motivation for word hashing wouldn't be as intuitive as in English. Thus, I choose to tackle the problem of unequal lengths in another way: by manually setting a maximum length parameter (default=20), longer texts would be cut short, and shorter texts would be padded with a sequence of null characters.

Word vector filtering. I use pre-trained Chinese word vectors [4] in the model. To accelerate the loading process during training, the word vectors are filtered beforehand, i.e. only words that occur in the training or test sets remain.

2.3 Deep Structured Semantic Models (DSSM)

As mentioned above, the architecture that I will adapt, DSSM, is first published by Microsoft Research in 2013. It is a deep semantic model, initially designed for information retrieval in search engines, i.e. given a query, the model retrieves the most relevant documents on the Web. The core concept of DSSM is that the network projects a query and candidate documents into a common low-dimensional vector space, where the semantic relatedness is computed as vector similarity scores. The model later evolves in other semantic mapping tasks, including graphic matching, captioning, Sent2Vec, etc. The last layer of the original network was to compute the cosine distance between two vectors, because it was originally for similarity computation. But here we have a binary classification task, so the objective function and the loss function should be adjusted.

[4] https://github.com/Embedding/Chinese-Word-Vectors

2.4 Convolutional Deep Structured Semantic Models (CDSSM)

Also brought forth by Microsoft Research, CDSSM has the following motivation: in the vanilla DSSM, texts are represented as bag of words, so the positional information within sentences are discarded. In CDSSM, the authors intend to capture the relation between words in a sentence by introducing convolution. The initial layers are the same as DSSM, but later a convolutional layer is followed by a max pooling layer, in order to discover salient contextual features. At last, a similar non-linear transformation is applied. It is shown that CDSSM significantly outperforms DSSM in information retrieval. Thus, I will explore both structures in the following section.

3 Experimental Results

3.1 Baseline

3.1.1 Model Architecture

Based on Huang (2013)[1], I re-implemented a slightly different DSSM as baseline, whose architecture diagram is as follows:

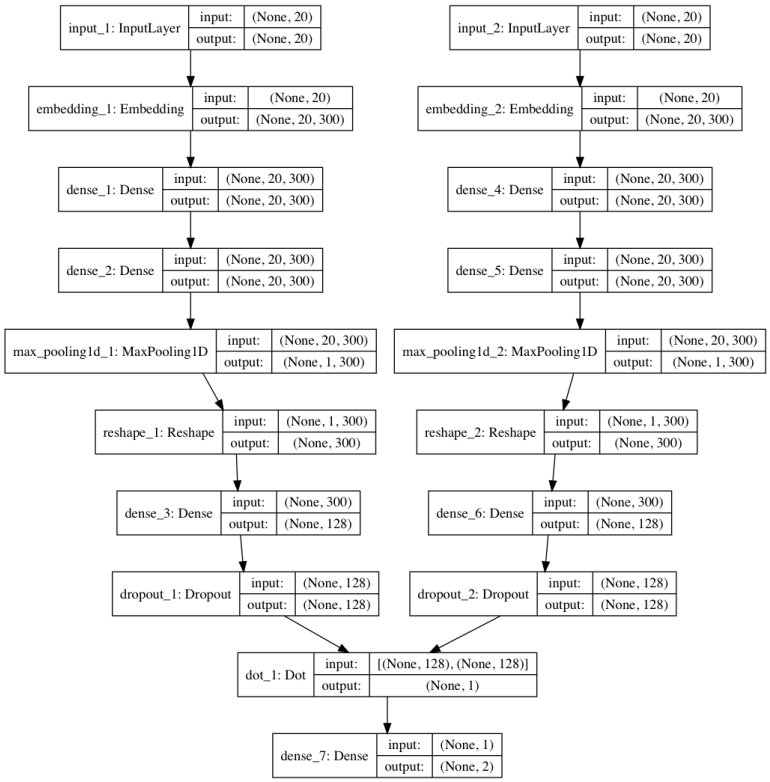

Figure 2: Baseline Model Architecture (based on DSSM)

The two texts in each pair are fed to the network as input_1 and input_2 respectively. Because the default maxlen parameter is set to 20, the input length here is limited to 20. The Input Layer is followed by the Embedding layer, where self-learned word embeddings instead of pre-trained word embeddings are used for the time being. Next are two fully-connected layers, similar to the Microsoft paper. A slight difference is that because there is no word hashing before, the dimension reduction is done by a pooling layer. Finally, the high-dimensional vectors are projected into an 128-dimensional space to produce the semantic representations for sentences. The cosine similarity between them is computed, and then we use Softmax to transform the similarity score to a binary classification probability, which is evaluated against the true label with the cross-entropy loss.

3.1.2 Several clarifications

Special character filtering. For the non-Chinese characters (e.g. mathematical symbols) in the corpus, the baseline model does not filter them out. The motivation is that if pre-trained word vectors are used, filtering is likely to

be better since the pre-trained word vectors almost do not include any mathematical symbols. But since the baseline model learns the word representations itself, the embeddings of such mathematical symbols could technically also be learned. So, I start with no filtering here.

Dropout layer. In the model structure diagram (Figure 2), there is a Dropout Layer, which does not exist in the Microsoft paper. However, for the baseline model, I temporarily set the dropout ratio parameter to 0, so the overall architecture is still mostly equivalent to the original paper.

Relevant parameters. maxlen=20, l2 regularization ratio=0.01, learning rate=0.001, embedding dim=300, train val split = 0.1, batch size=200, epoch=30, patience for early stopping=10.

3.1.3 Model performance

During the training process, 10% of the training data is used for validation. The loss and the accuracy in each epoch on the training and validation sets are plotted:

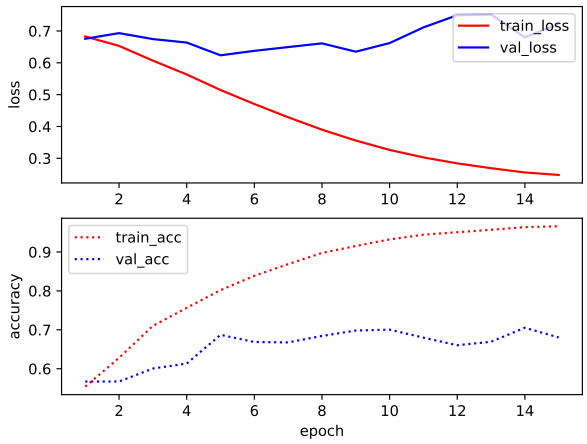

Figure 3: The performance of the baseline model during the training process

Obviously, overfitting is serious. Although the validation accuracy is slightly higher in the end than in the beginning, the overall trend fluctuates severely, and the validation loss has a tendency to rebound in the last few epochs.

On the test set, the performance of the final trained model is also evaluated (the final model is selected as the one with the lowest loss across all epochs):

Model	Accuracy	Precision	Recall	F1 score
Baseline	0.654	0.629	0.75	0.684

Table 1: The performance of the baseline model on the test set

Good news is that at least its performance is conspicuously above chance. Also, recall is much higher than precision, suggesting a tendency of over-predicting positive cases.

3.2 Extensions

On the baseline model, I experimented with several extensions from the following aspects:

3.2.1 Pre-processing and word embedding

In the baseline model, the special characters (e.g. mathematical symbols) in the texts are not filtered out. Here I experimented with two other filtering schemes, respectively named as "Coarse Filtering" and "Fine Filtering" :

"Coarse Filtering". All non-Chinese characters, including punctuation marks, are filtered out. It is expected that this scheme will be useful when using pre-trained word vectors, since they contain only Chinese characters with few exceptions. Another possibility is to use English word vectors at the same time, but the two semantic spaces then need to be projected into one shared bilingual space beforehand.

"Fine Filtering". Observe the sample problem text again:

```
两箱 水果 共重 $ $ 92 $ $ 千克 ， 第一 箱比 第二 箱重 $ $ 6 $ $ 千克 ， 两箱 水果 各重
多少 千克 ？ $ $ left （ 92 + 6 right ） div 2 = 49 $ $ （ 千克 ） $ $ cdotscdots
$ $ 第一 箱 ， $ $ 92 - 49 = 43 $ $ （ 千克 ） $ $ cdotscdots $ $ 第二 箱 .
```

Figure 4: A sample problem text from the dataset

In the Fine-Filtering scheme, aside from Chinese characters, I will try to retain special characters that contribute to semantic meaning, and it is expected will work better with self-learned word vectors for the same reason as above. Detailed filtering rules include the following:

- For LaTeX notations, only meaningful ones such as 'frac', 'sum', and 'pi' remain. On the contrary, notations such as 'end', 'begin', and 'equation' are mainly for formatting purposes, and thus have little semantic value.
- For maths operators, some (such as *, ^ , and +) are retained while others (such as -, /) are deleted. The motivation is that the first type has unambiguous semantic meaning, while the second, except as maths

operators, also appear as formatting notations (e.g. "/sum", " 二面角 B-AC-D").

- Numerical values are filtered out. For example in "x + y = 5", the number "5" is deleted. This is because numerical values are believed to have little influence on the conceptual meaning of problems. If two problem texts only differ in numerical values, they should still have high semantic similarity.
- For punctuation marks, only those that have clear semantic meaning (such as ? which indicates clear questioning) are kept. Other (such as commas and periods which are only for pause) are deleted.

With the above rules, the final cleaned problem text is:

```
两箱 水果 共重 千克 第一 箱比 第二 箱重 千克 两箱 水果 各重 多少
千克 ？ + div = 千克 cdot cdot 第一 箱 - = 千克 cdot cdot 第二 箱
```

Figure 5: The same sample problem text after Fine Filtering

It can be seen that the final text is free of semantically irrelevant information, and the meaning of the problem is almost completely preserved.

Using the same parameters as the baseline, the performance of the two models with new filtering schemes is as follows (Coarse Filtering is combined with pre-trained word vectors, Fine Filtering with self-learned vectors):

Model	Accuracy	Precision	Recall	F1 score
Baseline	0.654	0.629	**0.75**	0.684
Baseline+Coarse Filter	0.675	0.686	0.643	0.664
Baseline+Fine Filter	**0.717**	**0.707**	0.714	**0.710**

Table 2: The test performance of the new filtering schemes compared to baseline

Compared to the baseline model, the new filtering schemes both successfully improve test accuracy and precision. But the recall and F1 score for the Coarse Filtering scheme are lower than the baseline, which demonstrates that the overall performance of this scheme is not quite stable. It also indicates that the more characters are filtered out, the more conservative the model is when predicting positive samples. In comparison, despite a decrease in recall, the Fine Filtering scheme significantly improves F1 score along with accuracy at the same time, resulting in a well-balanced performance (all four metrics are around 0.70%). It may be safe to say that the Baseline+Fine Filtering model is the current optimal one, again suggesting the effectiveness of the carefully-designed filtering rules.

3.2.2 Dropout Layers

Based on the current optimal model in Section 3.2.1, another extension I try is to add a Dropout Layer (as shown in Figure 2, dropout ratio=0.1) before

calculating the cosine similarity, in order to alleviate overfitting. The results are presented in the table below:

Model	Accuracy	Precision	Recall	F1 score
Baseline+Fine Filter (3.2.1)	0.717	**0.707**	0.714	0.710
Baseline+Fine Filter+Dropout	**0.719**	0.698	**0.740**	**0.718**

Table 3: The test performance of the current optimal model (3.2.1) plus dropout

It is seen that after adding the Dropout Layer, precision falls by 0.09% yet recall jumps by 2.6%, which jointly improves accuracy and F1 score slightly. Generally speaking, the Baseline+Fine Filtering+Dropout model could be regarded as the current optimal.

3.2.3 Parameter tuning

Observing the training process of the above models, the problem of overfitting is still salient: in most cases, there is a bottleneck for the validation loss at around the 10th epoch, accompanied by slight fluctuations afterwards. But the training loss still drops steadily, often with early stopping. Therefore, based on the current optimal model in 3.2.2, here I try to tune certain parameters that may affect overfitting in this range:

- l2:[0.01, 0.05]
- drop out ratio:[0.0, 0.1, 0.2]
- batch size:[100, 200, 400]
- maxlen:[10, 20, 30]

Experiments show that the performance of the optimal model in the above range is as follows:

Model	Accuracy	Precision	Recall	F1 score
Baseline+Fine Filter+Dropout	0.719	0.698	0.740	0.718
Baseline+Fine Filter+Dropout+Param Tune	**0.723**	**0.704**	**0.740**	**0.721**

Table 4: The test performance of the current optimal model (3.2.2) plus parameter tuning

All metrics except recall are improved on the basis of the optimal model in 3.2.2. The optimal parameters are found to be: l2=0.05, drop out ratio=0.1, batch size=400, maxlen=10. It is seen that increasing l2, dropout ratio and batch size does help alleviate overfitting. Something worth noticing is that for the maxlen parameter (which stands for the maximum length of input texts), the baseline value is set to 20 purely for the sake of training efficiency. It is

previously thought that 20 is already a quite low value, since the average length of training texts is much longer. But surprisingly, experiments show that 10 is even better, which seems to indicate that the semantic similarity of two problem texts could be more accurately determined by only taking into consideration the first 10 characters.

3.2.4 Architecture Variation

Besides DSSM, here I also experiment with two other model architectures.

CDSSM. Based on Shen (2014a)[2] and Shen (2014b)[3], I built the following CDSSM structure:

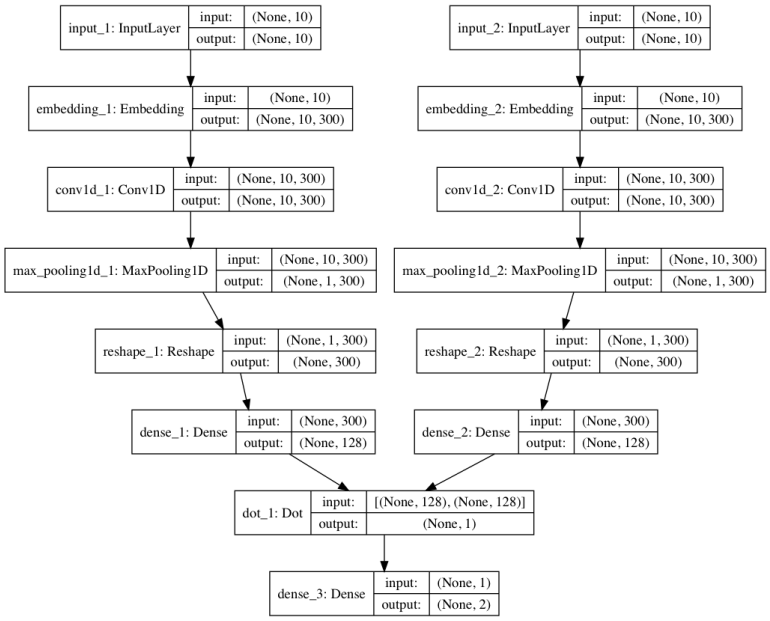

Figure 6: CDSSM Model Architecture

Similarly to the baseline model, the word hashing technique adopted by the original paper is not involved for processing Chinese texts. Additionally, since the current task is binary classification, the original parameter γ in the Softmax layer is no longer needed here.

The performance of the model on the test set is shown below:

Model	Accuracy	Precision	Recall	F1 score
CDSSM	0.685	0.681	0.694	0.687

Table 5: The performance of CDSSM on the test set

All parameters are the same as the optimal settings in Section 3.2.3, but the overall performance in terms of accuracy and F1 score is merely comparable to the baseline (3.1.3). It might be attributed to the fact that the optimal parameters are not transplantable, and another round of parameter tuning is needed. Because of the limit in time and resources, here I will skip that.

CGRU. One theoretical inadequacy of CDSSM is that the contextual information it captured is limited by the kernel size of convolution, and long-range dependencies are hardly preserved. Thus, here I adapt the previous CDSSM into a recurrent structure by adding an additional Convolutional Layer plus a Bidirectional GRU Layer, named as CGRU as a whole. The structure is shown below:

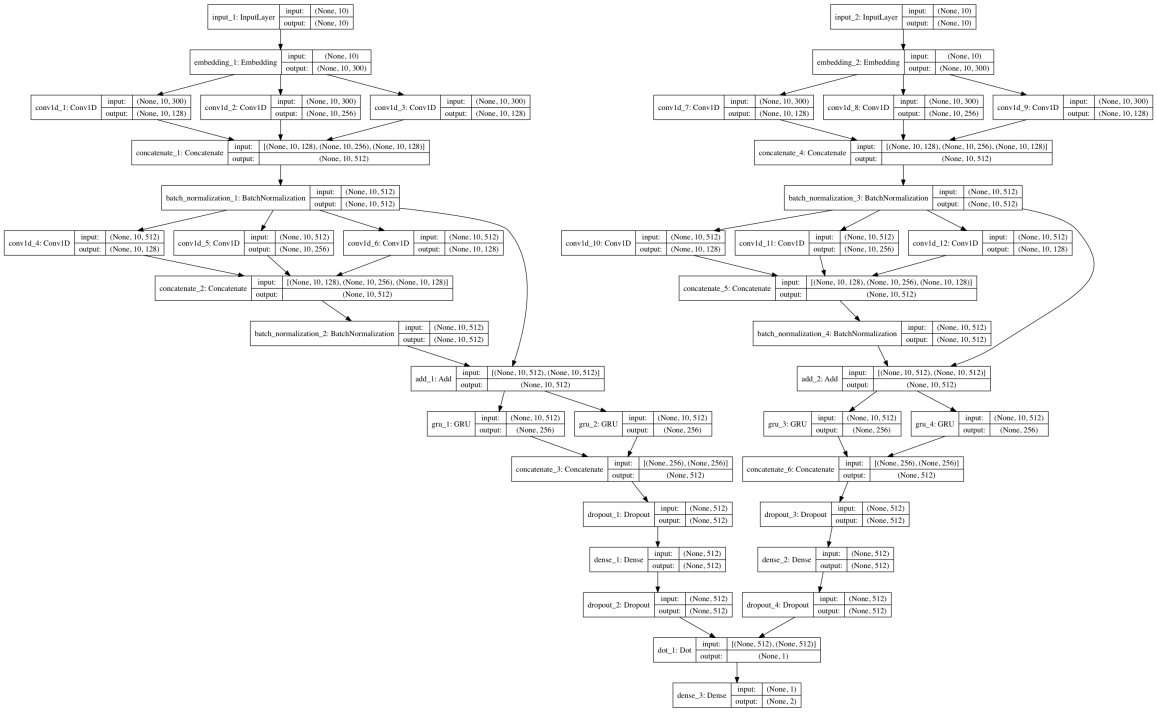

Figure 7: CGRU Model Architecture

On the test set, the performance of CGRU is:

Model	Accuracy	Precision	Recall	F1 score
CGRU	0.674	0.669	0.686	0.678

Table 6: The performance of CGRU on the test set

Unfortunately, CGRU even performs worse than simple CDSSM. Certainly, the similar lack of parameter tuning could be one possible explanation.

3.3 Text Retrieval: "10 most relevant problems"

Using the current optimal model (obtained in Section 3.2.3), here I will try to qualitatively demonstrate how the model captures semantic relatedness by drawing a few sample problems and retrieving the 10 most relevant problems from the dataset.

First, the problems in the dataset are deduplicated, resulting in a total of 19760 samples, each tagged with an ID. The three sample "queries" I select are the ones with ID 50, 100 and 150. Each of them (copied to make a list of length 19760) is fed into the model as the left input, and the rest "docs" (complied into a list) are fed as the right input. Running the prediction process yields 19760 float numbers, each standing for the relevance between the "query" and one "doc". The "10 most relevant problems" for the three sample problems are shown below:

- ID=50, problem text:

> *小明放学回家写完作业后想出门和小朋友一起玩，妈妈允许他最多出去半个小时，否则就要挨罚，他出门时是 5 时 14 分，那么小明最晚几时几分回到家才不会被妈妈罚？5 时 14 分+30 分=5 时 44 分，小明最晚 5 时 44 分回到家才不会被妈妈罚.*

10 most relevant problems:

ID	Similarity	Problem text
5809	1	小明用一个新本子写大字，第一天写了2页，第二天写了3页，第三天应从第？页开始写．小明用一个新本子写大字，第一天写了2页，第二天写了3页，第三天应从第6页开始写．
2017	1	小明家买了一箱牛奶，共15瓶，小明上个星期每个喝一瓶，还剩多少瓶？15-7=8（瓶）．
3526	1	18个同学排成一队做体操，从左边数小南是第6个，小南的右边有几个同学？18-6=12（个）．
7575	0.99999988	阿凡提在集市上花了600元买了一头小毛驴，转手以640元卖给了别人，随后又以650元买回了这头驴，第二天，阿凡提又以680元把驴卖了，那么阿凡提一共赚了元钱．第一天赚了40元，第二天赚了30元，一共赚了40+30=70元．
8299	0.99999973	张老师要带10个同学过河，已经有6个同学过了河，没有过河的还有几人？10-6+1=5（人）．
1497	0.99999952	一年级有120位同学去参观动物园，每人至少带一种饮料，其中带可乐的有78人，带矿泉水的有64人，既带可乐又带矿泉水的有几人？78+64-120=22（人）．
10234	0.9999994	小亮买了一本故事书，一共20页．他第一天看了几页，第二天又看了几页后，数了数还剩6页看完，请问小亮这两天一共看了多少页书？20-6=14（页）．
2012	0.99999833	一些小朋友排好队来报数，正着数小明报七，倒着数小明报九，一共有多少个小朋友？7+9-1=15（个）．
8298	0.99999774	三个兄弟比个子，老大比老二高5厘米，老三比老二高2厘米．那么老大和老三相差几厘米？老大和老三相差5-2=3
4078	0.99999738	有四个人在晚上准备通过一座摇摇欲坠的小桥．此桥每次只能让2个人同时通过，否则桥会倒塌．过桥的人必须要用到手电筒，不然会一脚踏空．只有一个手电筒．4个人的行走速度不同：小强用1分钟就可以过桥，中强要2分钟，大强要5分钟，最慢的太强需要10分钟．17分钟后桥就要倒了．请问：4个人要用分钟才能全部安全过桥？小强和中强先过桥，用2分钟；再由小强把电筒送过去，用1分钟，现在由大强跟太强一起过桥，用10分钟，过去以后叫中强把电筒送给小强用2分钟，最后小强与中强一起过河再用2分钟，他们一起用时间：2+1+10+2+2=17(分钟)．正好在桥倒塌的时候全部过河．(时间最短过河的原则是：时间长的一起过，时间短的来回过．这样保证总的时间是最短的)．

- ID=100, problem text:

> *如图所示，将自然数有规律地填入方格表中，请问：null*

10 most relevant problems:

ID	Similarity	Problem text
14209	1	有一列数：3，6，8，8，4，2，ldots，从第三个数起，每个数都是它前面两个数乘积的个位数字，那么这一列数中第2012个数是．按照要求这个数列应该是：3，6，8，8，4，2，8，6，8，8，4，2，8，6，8，8，4，ldots，观察可知，去掉开始的3和6后是8，8，4，2，8，6的循环，即周期为6，(2012-2)/6=335，所以第2012个数是每个周期的最后一个数，即6．
19228	1	请你找出规律填数．1，2，4，6，7，10，，这一数列中，各数有大有小，如果仅从相邻两数的关系入手，不易观察．这时可以从相隔的几个数里找出规律，将一列数相隔分成两列数，分别寻找它们各自的变化规律．如图：所以括号内应填10，14．
9898	0.99999998	先找出图中规律，再填数．A=，B=．找规律：1times4+times5=20，2+times3+times6=36．
8128	0.99999998	将0~2017这2018个自然数排成一行（不一定按从大到小或从小到大的顺序排列），得到一个6962位数，那么数字串“2017”最多能出现次．数字串“2017”可能是由一个数单独构成，也可能是由两个数或三个数或四个数构成．我们仍然按出现的“2017”进行分类：①2017四位数本身，1个；②20和17，20结尾的数，若是两位数，则只有1个；若是三位数，则百位可以是1~9共9个；若是四位数，则千位只能是1，百位可以是0~9共10个，20结尾共1+9+10=20个．17开头的数，仅四位数就有10times10=100个，排除掉1720，也有100-1=99个，所以这类最多有20个．③201和7，201结尾的数，若是三位数，则只有1个；若是四位数，则只能是1201这1个，共1+1=2个．7开头的数，仅两位数就有10个．所以这类最多有2个．④2和0和17，中间这个数只能是0，所以这类最多出现1次．⑤20和1和7，中间这个数只能是1，所以这类最多出现1次．但由于20结尾的数在①中已全部使用，所以这类不再统计．⑥2和0和1和7，中间只能是0和1，所以这类最多出现1次．但由于单独的0在④中已经使用，所以这类不再统计．综上，最多出现1+20+2+1=24次，而且都可以达到．
18605	0.99999988	数一数，下图共有几个点？并且列出算式．通过观察发现，每一行是4个棋子，一共4行，那么可以列出如下算式：4times4=16（个）；4+4+4+4=16（个）；1+2+3+4+3+2+1=16（个）．（方法不唯一）
5876	0.99999985	如图，将从5开始的连续自然数按规律填入数表中，请问第3行第100列的数是．暂无解析
1108	0.99999976	说一说，填哪几个数字，商是一位数？填哪几个数字，商是两位数？商是一位数 商是两位数；
10255	0.99999967	找规律填数．（1）按箭头的方向每相邻两数之间相差3，所以后面应该填：16+3=19，19+3=22．（2）按箭头的方向后面的数在前一个数的基础上依次+1、+2、+3、+4、+5，所以最后一个数应该填11+5=16．
11345	0.99999964	填一填．将第一个算式替换到第二个算式中，即8+square=10，所以square=2，再代入第一个算式，即2+2+2+bigcirc+bigcirc=8，bigcirc+bigcirc=2，所以bigcirc=1．
11811	0.99999964	找出下面各数列的规律，然后填空．1，5，6，1，7，8，，3，8；1，9，25，49，，121；1，2，5，14，41，，365；1，1，1，3，5，9，，31，57；1，，27，64，125，216；类兔子数列，从第三数起，每个数等于前面两数之和且只保留个位数字；平方数列，依次为奇数的平方；相差为等比数列的数列；类兔子数列，从第四数起，每个数等于前面三数之和；立方数列；

- ID=150, problem text:

*甲、乙、丙、丁四人拿出同样多的钱，一起订购同样规格的若干件新年礼物，礼物买来后，甲、乙、丙分别比丁多拿了3，7，14 件礼物，最后结算时，乙付给了丁 14 元钱，并且乙没有付给甲钱．那么丙应该再付给丁元钱．设丁拿了 a 件礼物，则四人花同样的钱，每人可以拿到 a+(3+7+14)/(4)=a+6 件礼物，实际情况：丁少拿了 6 件，乙多拿了 1 件，给丁 14 元，则货物单价 14 元，丙多拿了 14-6=8 件，3 件给甲，5 件给丁，5*14=70 元.*

10 most relevant problems:

ID	Similarity	Problem text
11618	1	有三筐西瓜一共39个，红筐比粉筐少7个西瓜，红筐比绿筐少2个西瓜你知道三筐各有多少个西瓜吗？方法一：用三筐的西瓜总数，减去粉筐比红筐多的，减去绿筐比红筐多的，就能算出三个红筐是多少，再除以3，就能求出红筐的西瓜数了．列式：39-7-2=30（个），红筐：30div3=10（个），粉筐：10+7=17（个），绿筐：10+2=12（个）．方法二：用三筐的西瓜总数，加上红筐比粉筐少的，加上绿筐比粉筐少的，就能算出三个粉筐是多少，再除以3，就能求出粉筐的西瓜数了．列式：39+7+5=51（个），粉筐：51div3=17（个），红筐：17-7=10（个），绿筐：17-5=12（个）．方法三：用三筐的西瓜总数，加上红筐比绿筐少的，减去粉筐比绿筐多的，就能算出三个绿筐是多少，再除以3，就能求出绿筐的西瓜数了．列式：39+2-5=36（个），绿筐：36div3=12（个），红筐：12-2=10（个），粉筐：12+5=17（个）．答：红筐有10个，粉筐有17个，绿筐有12个．
7297	1	某建筑工地开工前运来一批砖，开工后每天运进相同数量的砖，如果调15个工人砌砖墙，14天可以把砖用完．如果派来20个工人，9天可以把砖用完．现在派若干名工人砌了6天后，调走6名工人，其余工人又工作4天才能砌完．问原来有工人来砌墙？开工前运进的砖相当于“原有草堆”，开工后每天运进相同的砖相当于“新生长的草”，工人砌砖相当于“牛在吃草”，所以设1名工人1天砌砖数量为1份，那么每天运来的砖为(15×1420×9)÷(149)=6份，原有砖的数量为(156)×14=126份．现在派若干名工人砌了6天后，调走6名工人，其余工人又工作4天才砌完，如果不调走6名工人，那么这些工人共砌10天可砌完126+6×10+6×4=210，所以原有工人210÷10=21名．
10821	1	一项工程，若请甲工程队单独做需4个月完成，每月要耗资9万元；若请乙工程队单独做此项工程需6个月完成，每月耗资5万元．⑴甲、乙两工程队每月完成的工程量分别占全部工程的frac{1}{4}，frac{1}{6}，那么甲、乙合作所需时间为：1div(frac{1}{4}+frac{1}{6})=2.4个月；甲，乙合作2.4个月所耗资金为：(9+5)times2.4=33.6(万元)．⑵甲工程队完成全部工作要耗资9times4=36万元，乙工程队完成全部工作要耗资5times6=30万元，乙工程队耗资较少，为了节省资金，应尽量请乙工程队来做，但是乙工程队无法单独在五个月内完成工程，所以还需要请甲工程队来帮助完成一部分工程．所以，在五个月内完成的最好方案为：乙工程做5个月，甲工程队做(1-{5}/{6})div{1}/{4}={2}/{3}个月．即：甲、乙两工程队合作frac{2}{3}个月后，乙工程队再单独做frac{1}{3}个月．
5195	0.99999874	一班、二班、三班各有不同数目的图书．如果一班拿出本班的一部分图书分给二班、三班，使这两个班的图书各增加一倍；然后二班也拿出一部分图书分给一班、三班，使这两个班的图书各增加一倍；接着三班也拿出一部分图书分给一班、二班，使这两个班的图书各增加一倍．这时，三个班的图书数目都是48本．求一班原来有图书本．我们可采用倒推法，再结合列举法进行分析推理．在每一次重新变化后，三个班的图书总数目是一个不变的数，由此，可从最后三个班的图书数目都是48本出发进行倒推，求每一次重新变化以前三个班各自的图书数目，逐步倒推出原有的图书数目．依据题意可知，一班、二班的图书数目各增加一倍才是48本，因此增加前各应有24本，所以一班、二班的图书数目各应减半，还给三班．其余各次，以此类推，把倒推解答的过程用箭头法表示：一班：二班：三班：一班最开始有78本书，二班有42本书，三班有24本书．
2413	0.99999858	艾迪有贺卡20张，薇儿有贺卡25张．艾迪给薇儿几张卡片后，薇儿的卡片张数就是艾迪的2倍？现在艾迪和薇儿共有贺卡(20+25)张，艾迪拿出几张贺卡给薇儿后，他们的贺卡总数还是45张．当薇儿的贺卡张数是艾迪的2倍时，艾迪有：45div(2+1)=15(张)，艾迪给了薇儿：20-15=5(张)．
12285	0.99999726	一家三口人，三人年龄之和是74岁，妈妈比爸爸小2岁，妈妈的年龄是儿子年龄的4倍．问：三人各是多少岁？方法一：根据题意可得：74-2=72是儿子年龄的1+4+4=9倍，即儿子年龄为72div9=8（岁），妈妈年龄为4times8=32（岁），爸爸年龄为4times8+2=34（岁）．方法二：设儿子年龄为x岁，则妈妈年龄为4x岁，爸爸年龄为4x+2岁，根据“三人年龄之和是74岁”，可列式子：x+4x+4x+2=74，解得x=8，即儿子年龄为8岁，妈妈年龄为4x=4times8=32岁，爸爸年龄为4x+2=4times8+2=34岁．
7341	0.99999509	甲、乙、丙、丁共有糖果17颗，且每人的糖果数都不超过9颗，他们有如下的对话：甲对乙说：“如果我给你1颗糖，我们的糖果数就相同了．”乙对甲说：“如果你给我2颗糖，我的糖果数就是你的3倍了．”丙对甲说：“如果我给你3颗糖，你的糖果数就是我的3倍了．”丁对甲说：“如果你给我4颗糖，我的糖果数就是你的4倍了．”结果发现：糖果数是奇数的人说的都是对的，而糖果数是偶数的人说的都是错的．设甲、乙、丙、丁依次拥有A、B、C、D颗，那么，四位数overline{ABCD}是．由已知，四个人的和是奇数，则四个人中有三个奇数或一个奇数，则四个人中有三个或一个人是对的，通过甲、乙的话可得，甲乙两人要么都对，要么都错．如果甲乙两人都错，则两个人都有偶数个，此时丙丁中有一个对、一个错，只能丙是错的，而丙有偶数个，丁有奇数个，此时判断丁有偶数个，矛盾，因此甲乙两人都对．甲乙两人都有奇数个，只能是甲3个，乙1个，进一步判断丙是对的，有奇数个，即5个，那么丁有17-3-1-5=8个，满足要求，那么ABCD=3158．
1721	0.99999326	两筐千克数相同的苹果，甲筐卖出了千克，乙筐卖出19千克后，甲筐余下的千克数是乙筐的5倍，则甲、乙两筐原来各有千克苹果．用下图表示它们的关系：设乙筐余下的千克数为1份，则甲筐余下的千克数为3份，甲、乙两筐余下的苹果相差3-1=2（份），原来甲、乙两筐苹果的千克数相同，甲筐卖出了千克，乙筐卖出19千克，说明甲筐比乙筐少卖出19-7=12（千克），也就是乙筐余下的苹果比甲筐少12千克，所以甲、乙两筐余下的差是12千克，所对应的份数差是2，从而可以求出1份及两筐苹果原来的重量，甲、乙两筐余下的苹果数相差19-7=12（千克），乙筐余下苹果的数是12div(3-1)=6（千克），甲、乙两筐原来各有苹果的数量6+19=25（千克）．
5967	0.99999166	爸爸、妈妈今年的年龄和是72岁，5年后，爸爸比妈妈大6岁，求爸爸、妈妈今年各多少岁．“5年后，爸爸比妈妈大6岁”，也就是他们的年龄差是6岁，这是一个不变的量，所以今年爸爸、妈妈的年龄差仍然是6岁．这样，原题意就是“爸爸、妈妈的年龄和是72岁，年龄差是6岁，求二人各多少岁”的和差问题．那么,这道题就很简单了，列式：left(72+6right)div2=39（岁）——爸爸，left(72-6right)div2=33（岁）——妈妈，关于年龄问题的两个要点：①两人年龄差是永远不变的量，②两人年龄之间的倍数关系，随着年龄的变化，也在发生变化．
17010	0.99998939	爸爸和小高一起搬砖头，原计划爸爸搬其中的一些，小高搬剩余的砖头．父子二人发现，如果爸爸帮小高搬10块，那么,爸爸所搬的砖头数是小高的5倍；如果小高帮爸爸搬10块，那么,爸爸所搬的砖头数是小高的2倍，问原计划爸爸搬多少块砖头，小高搬多少块砖头？由题意知，砖头的总数是不变量，设砖头的总数是6份．因爸爸帮小高搬10块后，小高搬的是1份，所以小高开始搬的比1份多10块．又小高帮爸爸搬10块后，小高搬的是2份，那么,小高开始搬的比2份少10块．如图所示：所以1份对应的就是10+10=20块．因此所有砖头一共有20times6=120块，原计划小高搬20+10=30块，爸爸搬120-30=90块．

Observing the results, we have the following findings:

- **Problem 50.** It is a simple numerical addition and subtraction problem. Except for Problem 1497 (set intersection) and Problem 4078 (optimization), other problems are quite relevant, involving only simple addition and subtraction. The model performs fairly on this problem.
- **Problem 100.** It is a natural number inference problem. 7 of 10 problems retrieved are considered relevant, while the rest 3 (Problem 18605, Problem 1108, and Problem 11345) are less pertinent. The length of the original problem text is relatively short, which might be a reason why the retrieval performance is sub-optimal.
- **Problem 150.** It is a real-life numerical problem involving addition, subtraction, multiplication and division, and the concept of taking the average. Since keywords like " 相等/倍数/一样" are salient, almost all retrieved problems are similar, except only Problem 7341, a truth-condition validation problem. But regardless of the problem type, it still involves the four mathematical operations. So, in terms of concept, it may be still seen as relevant. In all, the performance of the model is also satisfactory on this problem.

4 Further Thoughts

- Sadly, the two complex architectures that I have experimented with, CDSSM and CGRU, have not achieved expected performance. One possible reason, of course, may be the lack of parameter tuning. If time permits, it would be ideal to conduct a performance analysis on individual data points, so as to see if there are any pattern in the errors. Also, it would be possible to verify if the theoretical effect of the new architectures (e.g. CGRU capturing long-range dependencies) are actually realized.
- By reexamining the corpus, it is found that the tokenization, which is completed by the data provider, is not quite accurate. A considerable proportion of "nonsense" words like " 图中缺"“可求”“开平”in the tokenized texts might have led to a significant number of Out-of-Vocabulary (OOV) words in training. It might be sensible to use character-level embeddings instead of, or combined with word-level embeddings in this case.
- Again, in terms of the difference between English and Chinese models, the original paper implements word hashing in order to reduce the dimensionality of one-hot vectors for English words, but Chinese does not have such a problem, and therefore word hashing is unnecessary. As a side effect, we have to find another way to solve the problem of unequal input lengths in the model. The method I have used, setting a maximum length plus padding, might not be careful enough. Probably it will be better to add a Pooling Layer as in CDSSM.

References

[1] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338. ACM, 2013.

[2] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 373–374. ACM, 2014.

[3] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 101–110. ACM, 2014.