# Hierarchical text segmentation for topic modeling

**Bin Wang**
Stanford University
bwang4@stanford.edu

**Michael Percy**
Stanford University
mp81@stanford.edu

## Abstract

Though plenty of algorithms have been explored to extract topics from text corpus, little attention has been put on representing a document with set of topics with exact locations in the document. Also, current techniques normally focus on how to model topics from the target corpus with little attention paid on how to take advantage of existing external models. This project tries to address these problems. We firstly propose an algorithm that converts a document into a hierarchical topic tree and then tries to resolve the "20 news groups" classification task with a model trained on Wikipedia.

## 1 Introduction

The major goal of this project is to explore a novel topic extraction method in which various topics within a document are explicitly mapped to corresponding document segments. Topics may be nested, i.e. a topic may be composed of smaller topics. In addition to identifying this structure, another difference between our targeting method and previous topic extraction methods such as LDA is that the topics are predetermined by a large text corpus such as Wikipedia.

Our hypothesis is that there exists a universal set of topics in any language. Any document can be decomposed into a set of these universal topics. The structure and set of topics that characterize the document can then be used as features for later tasks such as classification.

Thus, finding a good model of topics, design a good algorithm to convert a document to topic trees and evaluate the proposed model and topic extraction algorithm will be the major goals of this project. GloVe model (Pennington et al., 2014), ESA model (Gabrilovich and Markovitch, 2007) constructed from Wikipedia corpora and LDA model extracted from Wikipedia corpora are the major models we have explored. Regarding the topic extraction algorithm, the selective search algorithm for image object identification was modified to fit in text context. Regarding the evaluation we simply done a classification task on 20 news group data.

The following paper is organized in the following way. Section 2 briefly go through the prior work that is related to our project. Section 3 briefly explains our approaches. Details regarding the different topic models, hierarchical topic extraction algorithm and classification methods applied to the extracted topic tree are explained there. Section 4 summaries the results that was obtained. It is quite pity that no remarkable improvement was obtained during this project. Actually due to the limitation of computational resources, full testing result on many models are not obtained. But attempts were made to draw meaningful conclusions were made. Section 5 explains future work to fully evaluate this proposal. Section 6 discusses and concludes the paper.

## 2 Prior work

Works that are related to includes various word and topic models and selective search algorithm for object in images.

Topics models explored in this project include GloVe model, LDA model and ESA model. GloVe models (Pennington et al., 2014) each word in a way such that word pairs with similar relation will have similar word vector distance. In this project, we use the average of each word vector to represent the overall topic distribution of the text. The dimensions of word vector is used as the topics. ESA model (Gabrilovich and Markovitch, 2007) takes each article in Wikipedia as a topic as models the topic with the article's tf-idf vector. Then the distances between any unseen document and each

topic (Wikipedia article) can be calculated and also be used as a topic representation of the article. It is worth mentioning that the topic number in this number is the article number in Wikipedia. Hence this model is computationally expensive and could not be directly evaluated. To address this problem, a K-mean clustering was run against the potential topics. And the centroids of each cluster were considered the new topic models. But this clustering itself seems to be too expensive for personal computers. Only poorly converged topics were tested.

LDA model (Blei et al., 2003) is an unsupervised method that builds a set of topic models from an existing corpus. It will turn a document into a vector of topic composition. And we can use this vector to represent any document. The good aspect of this method is that we will have smaller number of topics so that later work such as topic search and classification will be less computational expensive. But training a LDA model, especially with a proper size of topic numbers is extremely expensive. Only a model with 100 topics were trained and tested.

Selective Search (Uijlings et al., 2013) is a method to propose potential object locations it takes an initial segmentation and recursively merges the most similar segments and records the areas that shows up until the whole images are merged. There are lots of tricks to play with the initial segmentation proposal, similarity measure between adjacent regions etc. on images. But in this project, we will simply use sentence as the initial region proposal and the cosine distance between each segments as the similarity measure.

## 3 Approach

### 3.1 Topic Models

As mentioned above, GloVe, ESA, clustered ESA and LDA are the topic models that were experimented with in this project. The GloVe dataset used was the 6B document, 300 dimension one from the GloVe web site, which was trained on Wikipedia. ESA and LDA models were trained from a Wikipedia dump. But the ESA model is very computationally expensive and only a 100-means clustered version was tested. Similarly, training of LDA are also very expensive, only and LDA model with 100 topics were trained and evaluated. Due to the limited topic size, the results obtained in this project may not be fully representative for the full capacity of each model. But it is still a good attempt and may reflect the potential of these models.

The LDA model was trained with default gensim API. But since the query and calculation of ESA model is too computationally expensive, the default gensim implementation had to be altered.

In order to quickly query the ESA model, we had to build a reverse index. The ESA model, with 3.7 million documents and 200K terms in its dictionary, is too large to fix in main memory on a standard server. The gensim implementation shards this index, allowing for memory-mapping to swap the shards in and out as necessary on a single machine. With some significant effort, we were able to build a reverse index that allowed querying based on terms in our small document segments, enabling us to only query a subset of the documents. This approach sped up feature extraction at index query time by around 2 orders of magnitude. The changes for this implementation are currently on http://github.com/mpercy/gensim but will be submitted to the main project as a pull request.

It turns out that classifying a data set with 3.7M features is quite time consuming. In order to address this issue, we also implemented a k-means clustering approach (Hartigan and Wong, 1979) for the ESA model. This clustered similar topics by cosine similarity, and reduced the topic dimensionality from 3.7M down to 2000, 500, and 200 (these are the cluster sizes we tested).

### 3.2 Hierarchical Topic Detection

This algorithm for hierarchical topic detection is developed from the selective search algorithm for image object detection. The basic idea is to start from some initial segmentation and recursively merge the most similar segments until the whole document is merged. Any region that showed up during the process is a potential region for a topic to appear. The pseudo code is shown below:

```
def topicSearch(# input text
                doc,
                # convert text to vector
                feature_extractor,
                # similarity function
                similarity,
                # splits text into sentences
                splitter )

    # initial proposal of regions
    segments = splitter(doc)

    # record initial regions as each sentence
    regions = [(i, i+1) for all segments]
```

```
    # Similarity set is a list of similarities
    # between a segment and its next segment.
    similarityWithNext = zeros()

    # Initialize similarities.
    similarityWithNext =
    [similarity(segment, next segment)
                        for all segment]

    while not all segments are merged:
        # Merge the most similar region.
        # regions will be modified with
        # the merge
        mostSimilarIndex =
         getMostSimilar(regions)
        mergedRegion = merge(mostSimilarIndex)

        if mergedRegion is not None:
            # Add new region to
            # hypotheses locations.
            regions.append(mergedRegion)

    return (segments, regions)
```

The above algorithm can list all the regions that are likely to contain some topics, but the regions are not organized in a tree structure. Hence, we still need to parse the regions into a tree. The pseudo code is shown below:

```
def parseTree(regs, length):
    regions = dict()
    for start, end in regs:
        try:
            regions[start].append(end)
        except:
            regions[start] = [end]
    if length not in regions[0]:
        raise Error
    root = TopicTree((0, length))
    regions[0].remove(length)

    def findChildren(node):
        s, e = node.region
        if e == s+1:
            return
        nxt = s
        while nxt < e:
            nxte = max(regions[nxt])
            tmp = TopicTree((nxt, nxte))
            node.children.append(tmp)
            regions[nxt].remove(nxte)
            nxt = nxte

        for child in node.children:
            findChildren(child)
    findChildren(root)
    return root
```

### 3.3 Feature Construction

Since evaluation is a classification task, features for the classifiers need to be constructed from the hierarchical topic trees.

All the models we use represent a piece of text as a vector, where each element of the vector represents the possibility for the topic to be the text topic or the similarity between the topic and the text. Hence, the possibility for a topic to show up or the similarity between the topic and the text will be will be the features. Three types of features were tested in this project – topic representation of the whole document, highest possibility of each topic and highest possibility of topics in each layer.

The topic representation of the whole document is straightforward – we take the whole document as the only region and convert it into a topic vector and use it as an feature. This feature will be referred as flat feature set.

The highest possibility of each topic may be a bit confusing by its name. Since we have many regions in a document, each regions will have a topic vector. One topic may have a possibility of 0.1 to show up in one region but may have a possibility 0.2 to show up in another. In this feature set, the number of features is still the number of topics in the model, but the value for each feature will be the highest possibility for the model in all regions. This feature set will be referred as highest possibility feature set.

The highest possibility of topics in each layer extends the previous feature set by considering each layer in the topic tree of a document. Since the topics tree is organized in layers, each layer will have a number of regions. And thus for each layer we will get a "highest possibility of each topic" feature set. By concatenate the highest possibility of each topic feature set in each layer, a new feature set can be obtained. The number of features in this feature set is the number of topics in the model times the number of layers that were taken into consideration. Since deeper the layer is, less important it will be, the possibility of topics of each layer also decays with their depth. This feature set will be referred as top K layer feature set since it takes the top K layer into consideration.

When used with GloVe, the top-K feature approach takes an additive approach to constructing GloVe vectors from each segment. Then, to construct the final feature vector for a document, the vectors for each layer are concatenated and weighted before classification.

### 3.4 Classification

Logistic regression, Gaussian Naive Bayes classifier, multinomial Naive Bayes classier and Support Vector Machine are the classifiers that were tested

in this project. Logistic regression outperforms other classifiers under most situations. Multinomial Naive Bayes classifier fits the LDA model better.

## 4 Evaluation

This section discusses the evaluation of the proposed method. In general, the proposed method didn't out perform the state of art methods. Carefully tuned Naive Bayes classifier can achieve a accuracy around 87%, but our best accuracy is around 74% with GloVe model, top K layer feature set and logistic regression.

This limited performance may be caused by many reasons. Firstly, none of the model is properly trained except the GloVe model. The huge number of articles result in a huge feature space in the original ESA model and making both training and inference slow, which was not properly emphasized in the original paper. Solutions with k-means clustering on the topics, LDA on Wikipedia corpus to extract fewer topics were proposed late and, unfortunately, these tasks are also computational expensive. The clustered topics were not well converged and the LDA model has only 100 topics, which apparently too small for generic tasks.

Despite the limited performance that is achieved within this project, we still consider the proposed method successful. This will explained in the baseline sub-section.

### 4.1 Baseline

Though the 20news group classification task has well developed algorithms, we tend to set our baseline to be a LDA based classifier. This classifier firstly train and LDA model with the training data. With the trained LDA model, all the documents are turned into a vector of topic compositions. Then a Multinomial Naive Bayes classifier is trained with the training data and evaluated with the test data.

This method is chosen as the baseline since it fits the objective of the project – the comparison between model which is trained with external data and the model that is trained only with training data, and the comparison with non-hierarchical topic extraction and hierarchical topic extraction. The performance of the baseline is not very good. It's precision is 0.32393, recall is 0.28564, and F1 score is 0.26356.

Though the LDA baseline is not performing so well, it is better to keep in mind that the state of art accuracy varies from 0.762 to 0.8486 according to http://nlp.stanford.edu/wiki/Software/Classifier/20_Newsgroups and can be as high as 0.97 according to https://github.com/yassersouri/classify-text.

### 4.2 Topic Extraction Example

To present a straight forward example how the proposed topic search algorithm works, this subsection will give an example on how a document is parsed by our algorithm.

The original document is shown in Figure 1. The document is from 20news group, under the category "comp.os.ms-windows.misc" with a document number of 8514. The initial segmentation of the documentation is shown in Figure 1. Each segmentation point is marked with "[]" withe segmentation index in inside.

It is clear that the initial segmentation is not ideally segmenting the document at each sentence end. Sentences in list that is not end with period or period that is used as email deliminators are not properly dealt with.

The regions in each layer is shown in Figure 2. Each region is marked by the start and end segment number. Despite the inaccuracy of the initial segmentation, it can be seen that the region divisions are relatively reasonable. segment 0 to 7 are headers, 13 to 17 are presentation types, 17 is the information for accepted presentations, 18 to 25 are the abstract information, 25 to 29 are acceptance information etc. But it is also clear that there are some inaccuracies like division at 12 is apparently a mistake.

In general, the region divisions are considered acceptable.

### 4.3 Classification Results

The classification results for each classifier, model and feature extractor are shown in Table 1. The major part of parameter tuning process for GloVe model with top K layer feature extractor and logistic regression is shown in Table 2.

The ESA and LDA models were evaluated on a reduced-size data set, however relative to GloVe on reduced-size data sets they were far inferior (about 10%) to GloVe even at those levels. Clustered ESA was run on the full data set.

In short, the best-performing model from the various approaches that we tried was a model using GloVe vectors with a hierarchical feature set of

[0]From: lipman@oasys.[1]dt.[2]navy.[3]mil (Robert Lipman)
Subject: [4]CALL FOR PRESENTATIONS: Navy SciViz/VR Seminar

CALL FOR PRESENTATIONS

NAVY SCIENTIFIC VISUALIZATION AND VIRTUAL REALITY SEMINAR

[5]Tuesday, June 22, 1993

[6]Carderock Division, Naval Surface Warfare Center
(formerly the David Taylor Research Center)
Bethesda, Maryland

[7]SPONSOR: NESS (Navy Engineering Software System) is sponsoring a one-day Navy Scientific Visualization and Virtual Reality Seminar. [8]The purpose of the seminar is to present and exchange information for Navy-related scientific visualization and virtual reality programs, research, developments, and applications.

[9]PRESENTATIONS: Presentations are solicited on all aspects of Navy-related scientific visualization and virtual reality. [10]All current work, works-in-progress, and proposed work by Navy organizations will be considered. [11]Four types of presentations are available.

[12]1. [13]Regular presentation: 20-30 minutes in length
2. [14]Short presentation: 10 minutes in length
3. [15]Video presentation: a stand-alone videotape (author need not attend the seminar)
4. [16]Scientific visualization or virtual reality demonstration (BYOH)

[17]Accepted presentations will not be published in any proceedings, however, viewgraphs and other materials will be reproduced for seminar attendees.

[18]ABSTRACTS: Authors should submit a one page abstract and/or videotape to:

[19] Robert Lipman
Naval Surface Warfare Center, Carderock Division
Code 2042
Bethesda, Maryland 20084-5000

[20] VOICE (301) 227-3618; [21]FAX (301) 227-5753
E-MAIL lipman@oasys.[22]dt.[23]navy.[24]mil

[25]Authors should include the type of presentation, their affiliations, addresses, telephone and FAX numbers, and addresses. [26]Multi-author papers should designate one point of contact.

[27]DEADLINES: The abstact submission deadline is April 30, 1993.
[28]Notification of acceptance will be sent by May 14, 1993.
[29]Materials for reproduction must be received by June 1, 1993.

[30]For further information, contact Robert Lipman at the above address.

[31] PLEASE DISTRIBUTE AS WIDELY AS POSSIBLE, THANKS.

[32]Robert Lipman | Internet: lipman@oasys.[33]dt.[34]navy.[35]mil
David Taylor Model Basin - CDNSWC | or: lip@ocean.[36]dt.[37]navy.[38]mil
Computational Signatures and | Voicenet: (301) 227-3618
Structures Group, Code 2042 | Factsnet: (301) 227-5753
Bethesda, Maryland 20084-5000 | Phishnet: stockings@long.[39]legs

The sixth sick shiek's sixth sheep's sick.[40]

Figure 1: Original document with initial segmentation

```
(0,40);

(0,12); (12,40);

(0,8); (8,12); (12,13); (13,40);

(0,7); (7,8); (8,9); (9,12); (12,13); (13,25); (25,40);

(0,3); (3,7); (7,8); (8,9); (9,10); (10,12); (12,13); (13,17);
(17,25); (25,30); (30,40);

(0,2); (2,3); (3,4); (4,7); (7,8); (8,9); (9,10); (10,11);
(11,12); (12,13); (13,16);(16,17); (17,18); (18,25); (25,29);
(29,30); (30,31); (31,40);

(0,1); (1,2); (2,3); (3,4); (4,6); (6,7); (7,8); (8,9); (9,10);
(10,11);(11,12); (12,13);(13,15); (15,16); (16,17); (17,18);
(18,21); (21,25); (25,28); (28,29); (29,30); (30,31); (31,32);
(32,40);

(0,1); (1,2); (2,3); (3,4); (4,5); (5,6); (6,7); (7,8); (8,9);
(9,10); (10,11); (11,12); (12,13); (13,14); (14,15); (15,16);
(16,17); (17,18); (18,20); (20,21);(21,24); (24,25); (25,27);
(27,28); (28,29); (29,30); (30,31); (31,32); (32,36); (36,40);

(0,1); (1,2); (2,3); (3,4); (4,5); (5,6); (6,7); (7,8); (8,9);
(9,10); (10,11); (11,12); (12,13); (13,14); (14,15); (15,16);
(16,17); (17,18); (18,19); (19,20); (20,21); (21,23); (23,24);
(24,25); (25,26); (26,27); (27,28); (28,29); (29,30); (30,31);
(31,32); (32,35); (35,36); (36,39); (39,40);
```

Figure 2: Regions at each layer

Table 1: Performance Comparison Table

| Classifier | Feature Extractor | Topic Model | Precision | Recall | F1 |
|---|---|---|---|---|---|
| Logistic Regression | Flat feature | GloVe | 0.7332 | 0.7332 | 0.7324 |
| Logistic Regression | Max Topic feature | GloVe | 0.5999 | 0.5999 | 0.6 |
| Logistic Regression | Top K feature | GloVe | 0.7459 | 0.7459 | 0.7445 |
| Logistic Regression | Flat feature | LDA | 0.57 | 0.625 | 0.5602 |
| Logistic Regression | Top K feature | LDA | 0.6375 | 0.625 | 0.6016 |
| Multinomial Naive Bayes | Flat feature | LDA | 0.4458 | 0.475 | 0.4266 |
| Multinomial Naive Bayes | Top K feature | LDA | 0.5541 | 0.6 | 0.55 |
| Multinomial Naive Bayes | Flat feature | ESA | 0.4703 | 0.4444 | 0.4112 |
| Logistic Regression | Flat feature | Clustered ESA | 0.5259 | 0.5386 | 0.5196 |
| Logistic Regression | Top K feature | Clustered ESA | 0.5389 | 0.5597 | 0.5424 |
| Multinomial Naive Bayes | Top K feature | Clustered ESA | 0.5585 | 0.5280 | 0.5115 |

6 levels and a logistic regression classifier. With this approach we got a .7445 F1 score on our 10% test set. While varying the L2 C-regularization on the classifier did not help us get a better test error, our training F1 score was 0.8808, which leads us to believe that these features are highly predictive and with more tuning it may be possible to do even better. Our approaches using ESA and LDA had training error similar to the test error, leading us to believe that those features are less predictive than GloVe for this task.

## 5 Future Work

Since the GloVe model is the only properly trained model in current project, the major future work should be to test more deeply trained models. To be more specific, LDA model with proper topic number, say 1000 to 2000, should be trained and evaluated properly. Given time and compute resources, a larger clustering size of ESA could also be tested, to see if the predictive power of the features is increased. According to our experiment results, it can be seen that the LDA and ESA model does not generalize well. This may be caused by the fact that the model trained with Wikipedia may not have the most universal topics. Hence other topic representation models should be developed and tested. Thirdly, it can be seen that current initial segmentation cannot segment each sentences accurately, better initial segmentation proposals should be developed. In addition, our models were trained without stop words, without sense disambiguation etc which will certainly limit the classification accuracy. These features could be properly considered in the future. More efficient and reliable model training implementation should be provided since any model that is good enough for general tasks are meant to be large, complicated and hard to be trained. Thus such implementation is important. Last but not least, good combinations of this general topic model and task specific models should be explored. In current task, only the models trained from Wikipedia is used, which is not the best choice. Aid from a universal external model should be a help instead of a limitation.

## 6 Conclusions

In this paper, we reported our exploration on hierarchical topic extraction from a document with the aid of a external topic set. In the project, many innovative algorithms and attempts were made and the frame work of developing and utilizing such a external topic set has been set. But due to the time and resource limitation, most of the topic models are not well trained and the proposed algorithm didn't out perform other algorithms. But we consider our proposals – well trained external model can perform better than the model that is trained only the training data and hierarchical topics extraction is better than plain topics are well illustrated.

## References

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611.

John A Hartigan and Manchek A Wong. 1979. Algo-

rithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12.

Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. 2013. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.

Table 2: Major Part of Parameter Tuning Result for Top K Feature with GloVe Model

| Precision | Recall | F1 | Decay | Depth |
|---|---|---|---|---|
| 0.732335352959 | 0.736010493574 | 0.733222407099 | 0.2 | 0 |
| 0.73463297217 | 0.738600260082 | 0.73544093178 | 0.2 | 1 |
| 0.73277803756 | 0.736051441386 | 0.73377703827 | 0.2 | 2 |
| 0.727756676 | 0.730041764519 | 0.729339988907 | 0.2 | 3 |
| 0.735546545436 | 0.740117236306 | 0.735995562951 | 0.2 | 4 |
| 0.730630116667 | 0.733923604191 | 0.731003882418 | 0.2 | 5 |
| 0.729913238977 | 0.733255292961 | 0.730449251248 | 0.2 | 6 |
| 0.733132815214 | 0.736551114221 | 0.733222407099 | 0.2 | 7 |
| 0.732466481113 | 0.838856891743 | 0.735487522228 | 0.2 | 1 |
| 0.732335352959 | 0.736010493574 | 0.733222407099 | 0.4 | 0 |
| 0.736375948146 | 0.740046850576 | 0.737104825291 | 0.4 | 1 |
| 0.729948545241 | 0.732089658895 | 0.731003882418 | 0.4 | 2 |
| 0.73706685678 | 0.740327658192 | 0.737104825291 | 0.4 | 3 |
| 0.737217684584 | 0.740674758968 | 0.737659456461 | 0.4 | 4 |
| 0.738588658702 | 0.741902171116 | 0.738768718802 | 0.4 | 5 |
| 0.726896108818 | 0.729193340008 | 0.727676095397 | 0.4 | 6 |
| 0.728104976678 | 0.730499781032 | 0.729339988907 | 0.4 | 7 |
| 0.73677688129 | 0.847768268163 | 0.74013706396 | 0.4 | 1 |
| 0.727562084109 | 0.849072558382 | 0.728745360183 | 0.6 | 1 |
| 0.732178507423 | 0.737194892746 | 0.732667775929 | 0.8 | 3 |
| 0.731488266095 | 0.735185152571 | 0.732113144759 | 0.8 | 4 |
| 0.74454864911 | 0.74946188851 | 0.745978924016 | 0.8 | 5 |
| 0.740563586909 | 0.745997313576 | 0.742096505824 | 0.8 | 6 |
| 0.736667999093 | 0.741841768277 | 0.737659456461 | 0.8 | 7 |
| 0.735489094763 | 0.852836114383 | 0.738504045848 | 0.8 | 1 |
| 0.732335352959 | 0.736010493574 | 0.733222407099 | 0.95 | 0 |
| 0.733537402829 | 0.737154425243 | 0.73377703827 | 0.95 | 1 |
| 0.733024002182 | 0.736348150147 | 0.73377703827 | 0.95 | 2 |
| 0.734128453053 | 0.737758810867 | 0.73488630061 | 0.95 | 3 |
| 0.73995426118 | 0.744575493711 | 0.741541874653 | 0.95 | 4 |
| 0.736281085729 | 0.740451518458 | 0.738214087632 | 0.95 | 5 |
| 0.734158727489 | 0.737950542664 | 0.73544093178 | 0.95 | 6 |
| 0.732021328417 | 0.740937350116 | 0.730449251248 | 0.95 | 7 |
| 0.732087423335 | 0.853752902044 | 0.735482414609 | 0.95 | 1 |
| 0.732335352959 | 0.736010493574 | 0.733222407099 | 0.9 | 0 |
| 0.735515819681 | 0.739600757073 | 0.735995562951 | 0.9 | 1 |
| 0.735287563536 | 0.739389082944 | 0.735995562951 | 0.9 | 2 |
| 0.734831516758 | 0.737881259359 | 0.735995562951 | 0.9 | 3 |
| 0.73884389678 | 0.743871353838 | 0.739877981143 | 0.9 | 4 |
| 0.74071794524 | 0.74584284112 | 0.742096505824 | 0.9 | 5 |
| 0.739971538335 | 0.74417699505 | 0.741541874653 | 0.9 | 6 |
| 0.734820163328 | 0.745355410346 | 0.73377703827 | 0.9 | 7 |
| 0.735981573782 | 0.851377375124 | 0.73932033532 | 0.9 | 1 |
| 0.732335352959 | 0.736010493574 | 0.733222407099 | 1.0 | 0 |
| 0.728838700245 | 0.731290119678 | 0.729339988907 | 1.0 | 1 |
| 0.724388612994 | 0.728011707262 | 0.724902939545 | 1.0 | 2 |