UNIVERSITY OF AMSTERDAM

MSc ARTIFICIAL INTELLIGENCE
MASTER THESIS

Question Retrieval in Community Question Answering Enhanced by Tags Information in a Deep Neural Network Framework

by

CHRISTINA ZAVOU

1146273

December 18, 2017

36 ECTs
01-June - 18-December 2017

Supervisor:
Dr MAARTEN DE RIJKE

Assessor:
Dr EFSTRATIOS GAVVES

FACULTY OF SCIENCE

Acknowledgements

Firstly, I would like to thank Dr. Maarten de Rijke for his help and guidance throughout the duration of this thesis. The enlightening comments and feedback i received in our discussions helped me not to lose sight of the big picture, while struggling with particular details of the thesis.

Also, I would like to give special thanks to Nikos Voskarides for all the insightful comments and continuous feedback I received for my various and frequently questions.

Thanks also to my fellow master students Andy and George, with whose I became good friends and worked on common projects during this master degree, as well as George, my family and friends for all their support these couple of years away from home.

Finally, I thank Dr. Maarten de Rijke and Dr. Efstratios Gavves for agreeing to be members of the examination committee.

Abstract

Community Question Answering (CQA) platforms need to be easy and fast in question or answer exploration. It is common to use tags to categorize items in these platforms, and create taxonomies that assist exploration, indexing and searching. The focus of this thesis lies in recommending similar questions (Question Retrieval) by simultaneously deciding whether the contexts of two questions are similar, and which tags are applicable for each question. Current methods targeted for Question Retrieval in CQA either consider deep learning approaches [44, 8], or conventional approaches that utilize the available information on questions' tags [11, 83]. The former framework is proved to be more powerful –especially in the case with loads of available data–, while the later is faster and successful in cases with few data. In this thesis, a deep learning approach for both question retrieval and tags recommendation is proposed, and their joint learning is found successful for transferring knowledge in the Question Retrieval task, after applying it on the AskUbuntu forum data. Additionally the neural network based Tag Recommendation performs better than the existing conventional methods.

Contents

1 Introduction **5**
1.1 Research Questions . . . 8
1.2 Contributions . . . 8
1.3 Thesis Outline . . . 8

2 Background Knowledge **9**
2.1 Feed Forward Neural Networks . . . 9
2.2 Training Neural Networks . . . 9
2.2.1 Backpropagation . . . 9
2.2.2 Regularization . . . 10
Dropout Technique . . . 10
Early Stopping . . . 10
Weight penalty . . . 10
2.3 Convolutional Neural Networks . . . 10
2.4 Recurrent Neural Networks . . . 11
Bidirectional RNNs . . . 12

3 Related Work **14**
3.1 Question Retrieval . . . 14
3.1.1 Conventional Approaches . . . 14
Okapi BM25 . . . 15
3.1.2 Neural Network Approaches . . . 15
3.2 Multi Label Classification . . . 16
3.3 Tags Recommendation . . . 17
3.4 Multi Task Learning . . . 18

4 Problem Definition and Methodology **20**
4.1 Question Retrieval Model . . . 21
4.1.1 From Input Representation to the Encoded Representation . . . 21
4.1.2 Training the Encoder . . . 22
4.1.3 Inference . . . 23
4.2 Tags Recommendation Model . . . 23
4.3 Embedding the Tags Recommendation task within the Question Retrieval problem . . . 24
4.3.1 Question Retrieval model pretrained on Tags Recommendation task . . . 24
4.3.2 Joint Training of Question Retrieval and Tags Recommendation tasks . . . 24

5 Experimental Setup **26**
5.1 Data . . . 26
5.1.1 Data Sets for Question Retrieval . . . 26
5.1.2 Data Sets for Tags Recommendation . . . 27
5.2 Pre-trained word embeddings . . . 27
5.3 Implementation Details . . . 27
5.4 Baselines . . . 28
5.4.1 Question Retrieval Baselines . . . 28
5.4.2 Tags Recommendation Baselines . . . 28
TagCombine . . . 28
Binary Relevance Logistic Regression . . . 28
5.5 Evaluation Metrics . . . 29
Precision at Rank k (P@k) . . . 29
Recall at rank k (R@k) . . . 29
Mean Average Precision (MAP) . . . 29
Mean Reciprocal Rank (MRR) . . . 30
F1-score at rank k (F1@k) . . . 30
5.5.1 Statistical Significance . . . 30
Multiple Comparisons Problem . . . 30

6 Results and Discussion **31**
6.1 Neural Network-based Question Retrieval Model 31
Neural Network based ranker compared to BM25 model 32
6.1.1 Evidence supporting the tags utilization 32
6.2 Neural Network Based Tags Recommendation Model 35
NN-based model versus TagCombine 36
NN-based model performance 36
6.3 Question Retrieval Model pre-trained on the Tags Prediction task 36
6.4 Question Retrieval and Tags Prediction jointly trained 37

7 Conclusions and Future Work **40**
7.1 Main Conclusions 40
7.2 Recommendations for future work 41

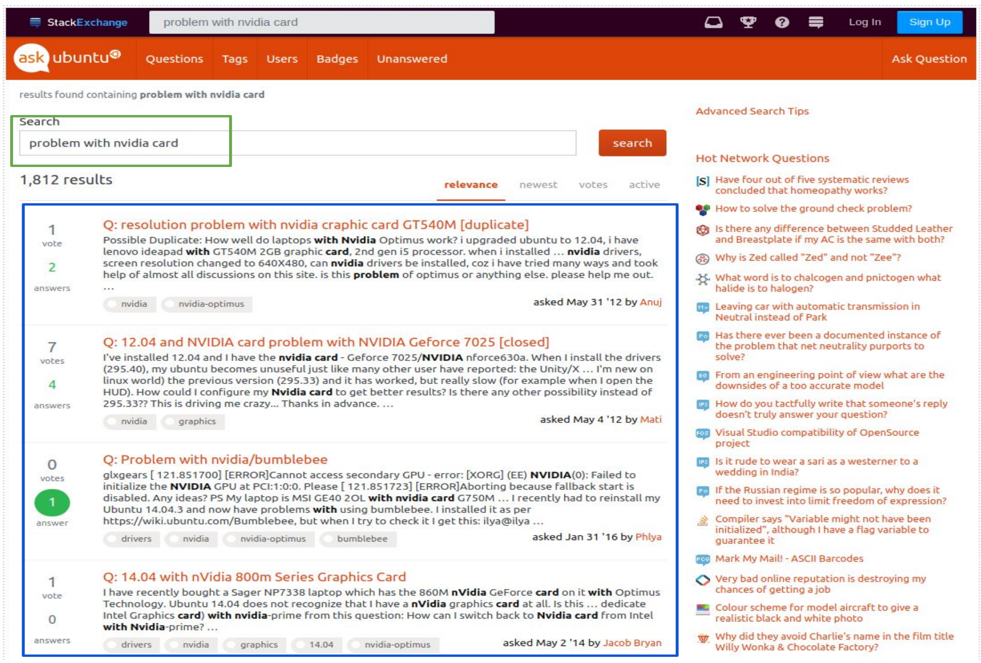
A Appendix **47**

B Appendix **55**

1 Introduction

Today, web users communicating their issues through on-line platforms is a very common behavior and data stored in online databases are uncountable. Machine Learning is an area that has shown its power and value in dealing with lots of data, and Information Retrieval (IR) and Natural Language Processing (NLP) areas have seen great advances from its use; from extracting important information out of data, to understanding and generating language like translations and summaries.

As the data grows, the indexing and searching tools need to keep high efficiency both in time and retrieved results. This makes Community Question Answering (CQA) forums an attractive area for the Information Retrieval research community. One of the main functionalities forming a CQA forum is Question Retrieval (QR); see figure 1. Given a query question and all the stored questions in a database, the goal of a question retrieval system is to rank all the questions and return the top similar to the query. Conventional approaches like the BM25 model, are commonly used in QR due to their fast and easy implementation. However, Deep Learning techniques promise more robust results and better performance on this task [44, 61, 8, 67], enhancing the perspective that Deep Learning is here to stay.



results found containing **problem with nvidia card**

Search: problem with nvidia card

1,812 results

1 vote, 2 answers
Q: resolution problem with nvidia craphic card GT540M [duplicate]
Possible Duplicate: How well do laptops **with Nvidia** Optimus work? i upgraded ubuntu to 12.04, i have lenovo ideapad **with** GT540M 2GB graphic **card**, 2nd gen i5 processor. when i installed … **nvidia** drivers, screen resolution changed to 640X480, can **nvidia** drivers be installed, coz i have tried many ways and took help of almost all discussions on this site. is this **problem** of optimus or anything else. please help me out. …
nvidia · nvidia-optimus — asked May 31 '12 by Anuj

7 votes, 4 answers
Q: 12.04 and NVIDIA card problem with NVIDIA Geforce 7025 [closed]
I've installed 12.04 and I have the **nvidia card** - Geforce 7025/**NVIDIA** nforce630a. When I install the drivers (295.40), my ubuntu becomes unuseful just like many other user have reported: the Unity/X … I'm new on linux world) the previous version (295.33) and it has worked, but really slow (for example when I open the HUD). How could I configure my **Nvidia card** to get better results? Is there any other possibility instead of 295.33?? This is driving me crazy... Thanks in advance. …
nvidia · graphics — asked May 4 '12 by Mati

0 votes, 1 answer
Q: Problem with nvidia/bumblebee
glxgears [121.851700] [ERROR]Cannot access secondary GPU - error: [XORG] (EE) **NVIDIA**(0): Failed to initialize the **NVIDIA** GPU at PCI:1:0:0. Please [121.851723] [ERROR]Aborting because fallback start is disabled. Any ideas? PS My laptop is MSI GE40 2OL **with nvidia card** G750M … I recently had to reinstall my Ubuntu 14.04.3 and now have problems **with** using bumblebee. I installed it as per https://wiki.ubuntu.com/Bumblebee, but when I try to check it I get this: ilya@ilya …
drivers · nvidia · nvidia-optimus · bumblebee — asked Jan 31 '16 by Phlya

1 vote, 0 answers
Q: 14.04 with nVidia 800m Series Graphics Card
I have recently bought a Sager NP7338 laptop which has the 860M **nVidia** GeForce **card** on it **with** Optimus Technology. Ubuntu 14.04 does not recognize that I have a **nVidia** graphics **card** at all. Is this … dedicate Intel Graphics **card**) **with nvidia**-prime from this question: How can I switch back to **Nvidia card** from Intel **with Nvidia**-prime? …
drivers · nvidia · graphics · 14.04 · nvidia-optimus — asked May 2 '14 by Jacob Bryan

Hot Network Questions
- Have four out of five systematic reviews concluded that homeopathy works?
- How to solve the ground check problem?
- Is there any difference between Studded Leather and Breastplate if my AC is the same with both?
- Why is Zed called "Zed" and not "Zee"?
- What word is to chalcogen and pnictogen what halide is to halogen?
- Leaving car with automatic transmission in Neutral instead of Park
- Has there ever been a documented instance of the problem that net neutrality purports to solve?
- From an engineering point of view what are the downsides of a too accurate model
- How do you tactfully write that someone's reply doesn't truly answer your question?
- Visual Studio compatibility of OpenSource project
- Is it rude to wear a sari as a westerner to a wedding in India?
- If the Russian regime is so popular, why does it need to invest into limit freedom of expression?
- Compiler says "Variable might not have been initialized", although I have a flag variable to guarantee it
- Mark My Mail! - ASCII Barcodes
- Very bad online reputation is destroying my chances of getting a job
- Colour scheme for model aircraft to give a realistic black and white photo
- Why did they avoid Charlie's name in the film title Willy Wonka & Chocolate Factory?

Fig. 1. Question Retrieval in AskUbuntu forum. The green box shows a query and the blue box shows the ranked list of retrieved questions.

The existing QR approaches usually face one common problem: the disambiguation. Disambiguation means that a part of text is not clear and is caused when a word can refer to multiple things –e.g. a pen can be a writing implement, or a female swan, and a screen can be referring to a laptop physical screen, or the software GNU Screen. In any case, more explanation is needed to define the exact topic mentioned. In the context of question retrieval in CQA, questions might refer to the same general topic, but on different specific cases, which require different solutions and pose the questions as dissimilar. The topic of this thesis lies in question retrieval via deep learning, focused on disambiguating the different topics under the same subject. More precisely, this thesis forms a research on the use of tag information that is usually available for forums' questions, to enhance the deep learning ranking model.

Similar research aiming to use the category metadata for improving the question retrieval task has been previously applied [37, 11, 83, 84]. In Cao et. al. 2010 [11], the authors enhanced question retrieval using the category information of questions; apart from the modeling of the probability of a question being similar to the query, they incorporated the probability of the question belonging to the query's category in order to reduce the set of candidate questions. In Zhou et. al. 2013 [83] the authors use the given category hierarchy of the forum, to propose a faster model which considers questions coming from similar categories (close in the hierarchy). In Hou et. al. 2015 [37] the tag information is used as an extra feature on their ensemble method to model question similarity. The more recent work in Zhou et. al. 2015 [84], lies closer to this research. The authors find a method to produce useful word embeddings to form questions' representations, on which the cosine similarity is enough to model question similarity. The mentioned works have used data from the Yahoo! Answers forum. In opposition, the present work is utilizing data from the AskUbuntu forum.

Tables 1 and 2 depict some results obtained by a Neural Network (NN)-based ranking model for the task of question retrieval in the AskUbuntu forum. It is straightforward to see that the tags of each question, are the main topics referenced or meant to refer to in the question context. Focusing on the example in table 1a, it is interesting to see the model being confused, when it retrieves –as the best candidate– a question which talks strictly about different issues with the ones talked in the query, and neither common text or common tags appear between the two. On the contrary, the truly similar object (golden truth), has one out of three tags in common. Similar trends are shown in table 1b. The retrieved question seems to be relevant at a first look, since both the query and the retrieved question talk about problems on watching youtube videos. If one is about to decide whether the retrieved or the similar question is more related to the query (which one can have similar solution), he might be unsure, but a look at their tags can be helpful to understand their difference, i.e. that the opera browser is an important similarity factor which is absent from the retrieved question.

Even though tags usually indicate the important topics addressed in a question, and as we have seen they can be indicative for question similarity, there are cases where the tags play no role in the similarity decision. Look for example table 2. In table 2a, we see similar questions and dissimilar questions both having one tag in common. Additionally, in table 2b we see the query and the incorrectly retrieved question having more common tags than the similar question and the query. This makes us question whether the tags can be a useful signal to give to the neural network ranker and motivates our research questions.

Query	what is a short cut i can use to switch applications ? in mac os it was command+tab . it is one of the most helpful tools , and i was wondering what was it 's ubuntu equivalent ? shortcut-keys, application-switcher
Retrieved	what does ubuntu use for getting/setting the time ? there is a way to change ubuntu 's system time in the gui date & time settings , however , as with most tools , i 'm assuming that is just a front-end for one of ubuntu 's command-line tools . what commands does ubuntu use to get or set the system time ? can these be used in bash scripts , or are they limited to only be executable by the system ? command-line, bash, time
Similar	keyboard short cut for switching between two or more instances of the same application ? i 'm wondering if there is a keyboard short cut for examining and switching between multiple windows of the same application ? i know of alt + tab but that only shows different applications . shortcut-keys, shortcuts, navigation

(a)

Query	ca n't watch youtube flash videos in opera browser on ubuntu i ca n't watch youtube videos in my opera browser on ubuntu 10.04 , the page is fully loaded but the space where the player is to be is blank (black) and i have flash player installed but it still wo n't work ... any advices on the matter ? flash, browser, opera
Retrieved	i ca n't watch youtube videos in either firefox or chromium . version 12.04 i ca n't watch youtube videos in either chromium browser or firefox . i 've tried installing a plugin or flash player , but neither seem to work . i 'm new to ubuntu and i 'm not real experienced on working on computers . 12.04
Similar	error in displaying youtube videos in opera i am using xubuntu 14.04 and opera browser . when i open youtube videos in it is displayed strangely . this question does not answer this problem.this error occurs only in opera browser . how do i fix this . xubuntu, opera

(b)

Table 1. Query questions from the AskUbuntu forum, with the first result retrieved by a neural ranking model and a true similar question to the query.

Query	how to disable networking from command line without sudo ? i want to disable networking from a bash script while not giving it administrative privileges . it 's possible from gui . is there a way to do it from cli ? networking, command-line, kde4
Retrieved	disable sudo permission to user from command line i have activated root user in ubuntu and want to use ubuntu as server with no de . for this i want to disable sudo privilege given to first user . how can i do this from command line ? i know i can use a gui but how to do it from command line ? command-line, sudo, users, privileges
Similar	how to resolve wifi disconnects and ca n't reconnect until disable and re-enable networking ? i do n't know if this is something that already has a solution or not . i am running ubuntu 12.04 lts and wireless driver rtl8187 . my wireless connection drops after a while (like an hour or so) and seems to occur after some period of inactivity from the mouse/keyboard . i have tried disabling the automatic screen shutdown and by association the suspend and this has not resolved the issue . in addition to the wireless connection dropping , it will not reconnect unless i disable and re-enable networking . i do this from the gui as i tend to have issues running network control commands via the command line if i run the network manager (which i like to use to monitor the network status via the tray icon when using the computer) . if there is any way to maintain the network connectivity i would be much obliged . thanks in advance , -jared 12.04, wireless, networking, network-manager, suspend

(a)

Query	can anyone tell me how to make guake terminal be part of the start-up applications ? how can i add guake terminal to the start-up applications ? command-line, startup-applications, guake
Retrieved	how to make scripts run in guake terminal instead of normal terminal ? i have installed guake terminal and i find it amazing . i have many scripts added as .desktop files in launcher . now i want these scripts to run in the guake terminal instead of opening in the normal gnome terminal . how can i achieve this ? the .desktop file is such : [desktop entry] type=application terminal=true icon=/path/to/icon/icon.svg name=app-name exec=/path/to/file/mount-unmount.sh name=app-name command-line, scripts, guake
Similar	how to make guake start by starup ? i 'm looking for help , because i had to install ubuntu one more time and i was always using guake terminal . in fact i do n't know how , but it was always starting when the desktop appeared and i only clicked f12 and did everything i had to . but now it does n't start automatically . i was looking for an answer but nothing worked . has anybody ideas how to solve it ? it would be really nice to use it as earlier ;) startup, guake

(b)

Table 2. Query questions from the AskUbuntu forum, with the first result retrieved by a neural ranking model and a true similar question to the query.

1.1 Research Questions

The main research question of this thesis regards an expert Neural Network model for the task of Question Retrieval in Community Question Answering forums. Precisely, we ask whether we can embed a model whose knowledge is the topics addressed in a forum's questions, into a model that is responsible for question retrieval, in order to make the latter an expert retrieval system.

In order to address this research question, we first generate a question retrieval model with neural networks. We refer to this us our main model for our main task. Then, we consider the following subquestions:

- Can a neural network based model be successful in recommending tags for questions (RQ1)?
- If such a model is feasible, we examine how it can be combined with our main model to give a better performance on the Question Retrieval task. For this, we consider following combinations:
  - Can we pretrain our main model on the tag recommendation task and get a better model for Question Retrieval (RQ2)?
  - Can we jointly train a model on question retrieval and tags recommendation and get better performance on our main task (RQ3)?

1.2 Contributions

The main contributions of this work to answer the research questions are:

- A neural network ranking model implemented with Tensorflow [1].
- Analysis and investigation of a neural ranking model compared to a lexical matching method (BM25), which shows that the main problem of the NN-based method is generalization. This makes our proposed method important for distinguishing general and specific problems addressed in similar questions.
- A neural network multi-label classification model for tags recommendation in the AskUbuntu [2] online forum, that beats the existing conventional approaches.
- Experimental analysis of different methods to transfer knowledge from the tags recommendation task to the question retrieval task, which shows that a NN-based model jointly trained on both tasks yields significant improvements on question retrieval.

1.3 Thesis Outline

The thesis is organized as follows. In Section 2 we provide background knowledge on Neural Networks. In Section 3 we provide work related to this thesis. Section 4 describes the methods used in our work, while Section 5 gives the experimental process, data and evaluation we used. In Section 6 all the results are reported and analyzed, and finally, conclusions and discussion about future research are given in Chapter 7.

[1] https://www.tensorflow.org/

[2] https://askubuntu.com/

2 Background Knowledge

2.1 Feed Forward Neural Networks

Artificial Neural Networks are computing systems inspired by the neurons of the brain, which receives signals and learn to recognize patterns in order to perform a task, like image recognition, voice recognition and other complicated tasks, for which the rule-based algorithms are infeasible. Such tasks are mainly divided into three categories, namely, Classification, Regression and Clusterization.

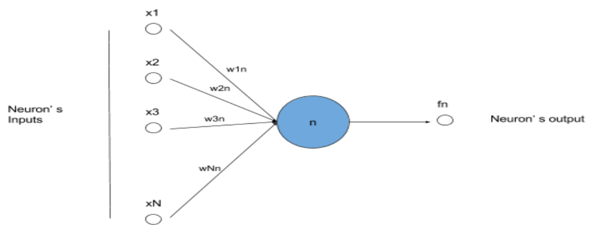


Fig. 2. An artificial neuron.

The neural networks are composed of multiple artificial neurons. A neuron 2 is a unit that receives multiple inputs, sums up the weighted values and uses an activation function to output a numeric value. In other words, a neuron is represented by the function f_n 1, where n denotes a specific neuron, a is an activation function, $\vec{w}$ is a vector of weights leading to the neuron and $\vec{x}$ is a vector of input values.

$$f_n = a(\sum_{i}^{N} w_{i_n} \cdot x_i) \tag{1}$$

When multiple neurons receive the same inputs, they form a neural network *layer*, namely a *Single Layer Perceptron*. If neurons in a layer send their outputs to other neurons, we say that multiple layers are *stacked*, creating a feed-forward neural network, namely a *Multi Layer Perceptron*. In other words, what distinguishes a Multi Layer Perceptron from a Single Layer Perceptron is that the former consists of an Input and an Output Layer, while the later has also an arbitrary amount of hidden layers.

The goal of a neural network is to find a mapping from the input data to the target data. Notice that the activation function is used to introduce non-linearities in the network. By only stacking multiple layers with multiple neurons, the network is only able to solve linearly separable problems, i.e. find linear mappings, since linear combinations are made on top of other linear combinations. when introducing an activation function, usually a log function, a sigmoid function, or a hyperbolic tangent, the network can transform these linear combinations into non-linear ones and solve hard problems.

2.2 Training Neural Networks

The training of neural networks regards the finding of a good estimation for the function that characterizes our target task. This implies the finding of parameters (weights in the network) that give the mapping equation from inputs to targets. To realize this learning process, a method called Backpropagation is usually utilized.

2.2.1 Backpropagation

Backpropagation is a gradient descent based technique. It has the precondition that a loss function is specified for a training instance and its goal is to minimize that loss. The loss function, composing of many free parameters (the weights), has a global minimum point, along with multiple local minima. The backpropagation algorithm aims to find the global minimum of the loss function, by repeating the following process.

- Firstly, complete a forward pass in the network, i.e. feed the network with a training instance and compute all neurons' output by passing preceding neurons' output to their successive neurons.
- Then calculate the loss signal and propagate it backwards in the network, by updating the all weights towards the direction of decrease in the error. For a vector of weights $\vec{w}$, the update will be $\nabla(\vec{w}) = -r\frac{\nabla L}{\nabla \vec{w}}$, where L is the loss function and r is a free parameter, the learning rate.

Traditionally, the training process was an online process, that is, one update of the parameters was made after feeding the network with one training instance and calculating its error. The online training however has some downsides. The frequent gradients calculations results to noisy signals, which can prevent the network from reaching the global minimum point of the loss, and can make the learning process slow as well. A batch training process was next introduced, that was faster and yielded better results. In the batch training, an update of the parameters is done after feeding the network with all training samples and calculating the error of all. This process has its downsides as well. It requires memory capabilities that fit the whole dataset, and it can also result to premature optimization due to the very stable gradient signal. Nowadays, the most common learning process, is the mini-batch training. According to the mini-batch learning, the dataset is split into multiple batches, and one update of the parameters is done after feeding a batch in the network and after calculating the batch loss. This technique is computationally more efficient that the batch learning, and is less time consuming than the online learning. It also avoids to fall into local optima by calculating gradients that are not too noisy and not to steady, but for this, a good batch size must be set on front.

2.2.2 Regularization

Weights of neurons become tuned for specific features during the training, providing a form of specialization. The neighboring neurons can become depended on this specialization, resulting in a network that overfits the training data. To avoid the overfitting of networks, the following methods are usually used.

Dropout Technique
If neurons are randomly removed from the network during the training process, then other neurons need to handle the representations that missing neurons are specialized for, and account for the predictions of the missing neurons as well. This is what the Dropout Technique proposed by Srivastava et. al. 2014 [71] does and forces neurons to be less focused on specific knowledge, and learn wider knowledge.

To implement the dropout technique, we select neurons randomly to be ignored during the learning process. This means that their contribution to the activation of successive neurons is removed from one forward pass and their weights do not get updated during that backward pass. As a result, the network becomes less sensitive to specific weights and achieves better generalization, reduces the possibility of overfitting the training data.

Early Stopping
Early stopping combats overfitting on training data by fine tuning the number of training steps or epochs. We use a data set different from the training and test sets, but representative for the test set, to track the model's performance. When the performance of the validation set starts decreasing we stop the learning process.

To implement the dropout technique, we save the model parameters at regular time intervals and stop the training after an amount of 'patient' epochs, where the model performance gets no improvements. Then we select the saved model with the best performance.

Weight penalty
The weight penalty is a regularization method used not only in NNs but in many machine learning methods. Assuming that a model with large weights is more complex than a model with small or fewer weights, and has increased probabilities to overfit the training data, the weight penalty increases the model's error when weights are big. We do this either penalizing the square values of weights (L2 norm) or the absolute values of weights (L1 norm).

In this work we use L2 penalty, as well as dropout and early stopping.

2.3 Convolutional Neural Networks

CNN-based approaches apply on problems with sparse interactions, in contrast to the traditional feed-forward NNs where each output is interactive with each input, and benefit in memory by sharing parameters (i.e. reusing the filters). CNNs started as powerful networks for tasks regarding visual problems, but have already expanded their use in other areas like NLP.

Using convolutional neural networks to obtain text representations has been frequently used with various structure-implementations. We give an illustration of the implementation used in this work in figure 3.

We have an input text sequence, $s = w_1, w_2, ..., w_{|s|}$, with $|s| = 4$, and $w_i \in V$, that is the vocabulary of all words. Word embeddings (W) are the distributional vectors $v \in R^{dx|V|}$ of words. For convenience and ease of lookup operations in W, words are mapped to integer indices $1, ..., |V|$. Now the input sequence s can be transformed into the sequence matrix $s_{mat} \in R^{|s|xd}$, forming an input of concatenated embeddings.

$$s_{mat} = \begin{bmatrix} - & v_s^1 & - \\ - & v_s^2 & - \\ - & ... & - \\ - & v_s^{|s|} & - \end{bmatrix}$$

The input s_{mat} is narrowly convolved with an arbitrary amount of filters of size $[window, d]$, where $window$ is the filter window, resulting into representations of $[|s| - window + 1, d]$ for every filter (see figure 3). At this stage, average or maximum pooling is applied to the neurons, leaving the network with $|s| - window + 1$ [3] nodes per filter. When the nodes' outputs are concatenated, the network outputs an encoded representation of the input text.

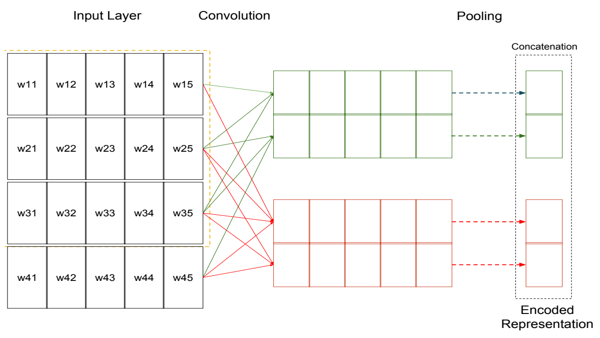


Fig. 3. Visual Representation of a convolution ($window = 3$) and a pooling layer that result to the encoded representation of a text sequence.

2.4 Recurrent Neural Networks

Countless learning tasks require dealing with sequential data and recurrent neural networks (RNNs) are powerful models that capture time dynamics via circular connections [47]. RNNs can process examples one at a time and retain a memory of arbitrarily long context window. Due to the modeling of time dependencies, this type of models became the most common approach when dealing with textual data, since language consists of structure and rules.

Fig. 4 depicts an artificial neuron with a recurrent edge, that is unrolled into a deep neural network with as many layers as the input sequence length (time steps). Because parameters are shared by all the time steps in the network, the gradient at each output depends not only on the calculations of the current time step, but also on the previous time steps. Therefore, *Back Propagation Through Time* must be employed.

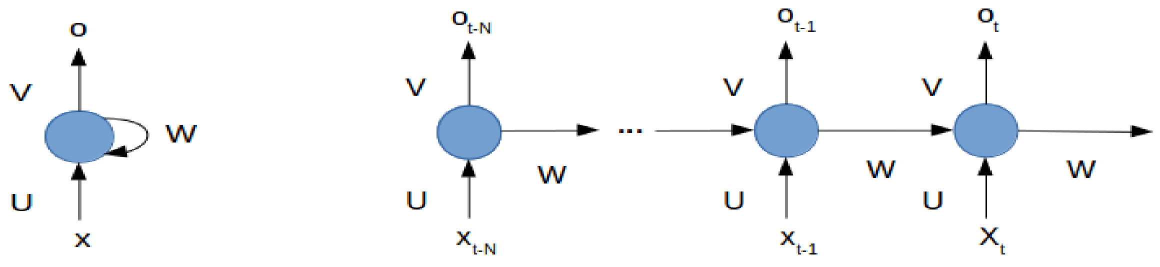


Fig. 4. A recurrent neuron (a) unrolled through time (b).

Training of recurrent neural networks is known as a hard problem, due to the learning of long-range dependencies [5]. The long-range dependencies result to vanishing or exploding gradients, that is, the negative

[3]Convolution output shape = $(sequence\_length - filter\_size + 2 * padding)/stride + 1$. Here padding is zero and stride is 1.

gradients propagated over very long sequences become too minimal over the sequence leading to stable neurons, or the positive gradients become too big leading to unstable neurons, correspondingly. In both cases, the neurons become incapable to learn.

These problems, are eliminated by modern recurrent architectures: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The LSTM model (initially proposed by [35]) replaces artificial neurons of a recurrent layer with memory cells. Memory cells consist of gates that are responsible for the degree to which new memory contents are added to their memory state, and when to update their state. Unlike traditional recurrent units which overwrite their content at each time-step, an LSTM unit is able to decide whether to keep the existing memory via the introduced gates and therefore it is possible to carry important information over a long distance, and able to capture long-distance dependencies [18]. Additionally, the LSTM unit ensures that no vanishing gradients will occur. Similarly, the GRU model (introduced in [17]) replaces artificial neurons of a recurrent layer with gated recurrent units. This unit has fewer gates than the LSTM one, but uses as well a mechanism to learn long-term dependencies. Because of the fewer parameters it is slightly faster to train and it is continuously gaining more attention.

Equations 2- 6 show how we map an input sequence $s = w_1, w_2, ..., w_{|s|}$, to a fixed-sized vector with an LSTM layer, that consists of an input gate, i_t, an output gate, o_t, a forget gate, f_t, and a memory cell, c_t, all of the same dimensions. The fixed-sized output vector is denoted by h_t. W^h is the recurrent connection from the previous hidden layer to the current hidden layer and W^i is the weight matrix connecting the inputs to the current hidden layer. Notice that the sigmoid functions squashes the vector values in $[0, 1]$ and then elementwise multiplications with previous state vectors define how much of the past knowledge we want to 'let in'; this is why they are called gates.

$$i_t = sigmoid(W_i^x \cdot w_t + W_i^h \cdot h_{t-1} + b_i) \tag{2}$$

$$f_t = sigmoid(W_f^x \cdot w_t + W_f^h \cdot h_{t-1} + b_f) \tag{3}$$

$$o_t = sigmoid(W_o^x \cdot w_t + W_o^h \cdot h_{t-1} + b_o) \tag{4}$$

$$logits_q = sigmoid(\vec{o}) \tag{5}$$

$$h_t = o_t \cdot tanh(c_t) \tag{6}$$

Similarly, equations 7- 9 show how we map the input sequence to the fixed-sized vector with a GRU layer, that consists of a reset gate, r an update gate, z and the output, h_t, all of the same dimensions.

$$z_t = sigmoid(W_z^x \cdot w_t + W_z^h \cdot h_{t-1} + b_z) \tag{7}$$

$$r_t = sigmoid(W_r^x \cdot w_t + W_r^h \cdot h_{t-1} + b_r) \tag{8}$$

$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot tanh(W_h^x \cdot w_t + W_h^h \cdot h_{t-1} + b_h) \tag{9}$$

Bidirectional RNNs

RNNs deploying either of these recurrent units have been shown to perform well in tasks that require capturing long-term dependencies, like the modeling of auto-encoders (seq2seq modeling) with the wide and popular applications in Machine Translation and Text Summarization [66]).

As depicted in fig. 5, a bi-directional RNN (first introduced in [65]) is build as two RNN models stacked on top of each other. It is a powerful model, that employs unlimited history and future information to make predictions, and has been applied to many tasks like parsing, translation and spoken language understanding [49, 33, 72].

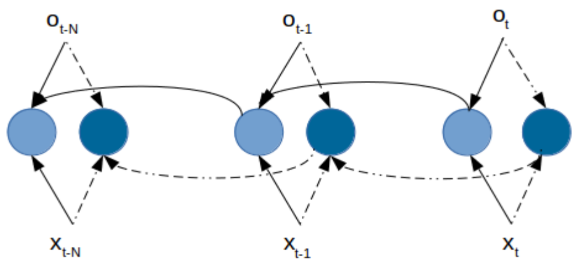


Fig. 5. Bi-directional recurrent units unfolded in 3 time steps.

3 Related Work

In this section an overview of work in various related research areas is provided, and more precisely in Question Retrieval, Tag Recommendation via Multi Label Classification, and Multi Task Learning.

3.1 Question Retrieval

The problem of question retrieval is not a new research area. By the time full-text search engines have been appeared (Google, Yahoo, Elasticsearch, etc.), this problem became an excellent challenge. The problem of question retrieval is closely related to areas like Semantic Search, Sentence Retrieval, Question Answering, Learning to Rank, Question recommendation, Community Question Answering, Question Duplicate Detection and Question Answering in general. Common scenarios where QR is needed:

- A Web user querying a question in the Web, retrieves some similar questions and skims the results until he finds the most representative to what he was looking for.
- A client support employee has been asked a question by some customer, then he retrieves the most similar questions in the database, and he manually finds the one that potentially answers the customer's question.

The examples do not end here. Community Question Answering (CQA) forums have become very famous, attracting more and more users to post their questions about anything they are concern with. Among the currently top forums, AskUbuntu is an example for Software developers, Quora is another example forum where everything is possible to be asked and Reddit is a social news aggregator forum. As the usage of these platforms increases, the need of powerful tools, able to retrieve possibly duplicate questions (example work in [8]) and possibly correct answers is a fact.

Question Retrieval has been approached by many researches in various ways. Goals of the research are to avoid the same issue being addressed twice, finding relevant questions that can help the user re-formalize his question, or finding relevant questions that partly answer his query. Moreover, finding similar questions can be an intermediate step in question answering. For example, in SemEval 2017 Task 3 [51] some researches were retrieving the best answer for a question, by first retrieving similar questions, and then selecting an answer among those questions' answers. The research for answer selection is even wider and longer; one recent deep learning approach is represented in [25].

Traditional question retrieval, and in general, question answering research, concentrates mostly on factoid questions. Factoid questions differ with open questions in that they are direct and rarely contain noise. The open questions posted on CQA forums are not grammatically correct, neither formal, and result in very noisy texts. Two similar questions posted on CQA forums can vary significantly in vocabulary, style, length and content quality [8], which makes the task of question-question (Q-Q) similarity a hard one due to the "lexical gap".

The main problem addressed in this work, is question retrieval for Software Information platforms like the AskUbuntu. By suggesting all relevant or same (paraphrased) questions to a new question that has arrived, one saves time for manually checking all the historical questions and their answers. The definition of the task follows:

Given a query question q, and M candidate questions q_1, q_2, q_3, ..., q_M, rank the candidates, with the most similar to q being on top and the least similar to q, being on the bottom of the list.

Some of the existing proposals model the users (based on their questions and answers) and enhance the system with this information. In this thesis, Question Retrieval is the main task and is addressed using the plain text (body, title and tags) to model question similarity.

3.1.1 Conventional Approaches

Most conventional approaches that address question retrieval begin with transforming the questions' text with the Bag-of-Words [30] representation.

Bag-of-Words representation: All (N) unique words extracted from the corpus define the vocabulary $V = \{w_1, w_2, w_3, ...w_N\}$. Then, a text, $t = "w_1 w_3 w_3 w_4 w_2"$ is transformed into the sparse vector [1, 1, 0, 1, 0, ..., 0]. In other words, every word of the vocabulary is represented by 0 in the multi-dimensional vector if it is absent from t, and 1 if it is present.

tf-idf weighting: Not all the words have the same importance. The tf-idf value is high for terms appearing often in the document (term frequency), but not in the whole corpus (inverse document frequency). The definition tf-idf for a word, w, and a question, d, in the collection of questions, D, is $tf(w,d) \cdot idf(w,D) = \frac{f_{wd}}{|d|} \cdot \frac{|D|}{|d \in D, w \in d|}$, with f denoting frequency.

Transforming the BoW representation by the tf-idf weighting results into the most common approach for text representation. The popular and conventional ranking method BM25 [58], uses the BoW representation with some weighted dependencies, and results into a simple, yet strong method 10. Another popular approach is Latent Dirichlet Allocation (LDA) [7]. LDA is a probabilistic model used to translate a collection of documents into a set of hidden context relations. It is a powerful and simple approach which has received a lot of attention from various research areas. In Question Retrieval, it is widely used to transform the questions into a set of latent topics that are useful for modeling questions' similarity. A very different state of the art conventional method is the use of a translation model that calculates the probabilities $p(q/q_x)$ and $p(q_x/q)$ to address the questions' similarity [82]. Most of the top methods in the Semeval 2017 task 3 compentition [51] are using a lot of feature engineering, exploiting kernel functions [26], tree kernel features from parse trees, as well as similarity features like cosine distance applied to lexical, syntactic, semantic or distributed representations.

Okapi BM25
Given a query question $q = w_q^1, w_q^2, w_q^3, ..., w_q^{|q|}$ and a candidate question $d = w_d^1, w_d^2, w_d^3, ...w_d^{|d|}$, which belong in the collection of questions D, the matching score of q and d is defined as

$$S(q,d) = \sum_i idf(w_q^i, D) \frac{tf(w_q^i, d)(k_1 + 1)}{tf(w_q^i, d) + k_1(1 - b + b\frac{|d|}{|\hat{d}|})} \tag{10}$$

with $|\hat{d}|$ denoting the average length of question in the collection, and k_1, b the free parameters of the model with default values $2, 0.75$, respectively.

3.1.2 Neural Network Approaches

A more sophisticated representation, that does not lose the ordering of the words and the semantics of text –like the previous methods do– is the word embeddings (Le and Mikolov [43]). A word is predicted as a vector representation by a neural network model that is trained to predict the next word given the context window of preceding words. The model is proved to learn the semantics of words, capturing them in the form of euclidean distances. Given the word embeddings, vector representations for an arbitrary length of text can be easily constructed (either by concatenation, summation, or by averaging). Le and Mikolov recently proposed a stronger method for text representation, namely, the paragraph vector [74], that represents a document into a fixed-length vector. The comparison of two documents (or two questions), can be then found by the cosine similarity of their vector representations.

Deep Learning techniques in text similarity has already shown potential application, by outperforming conventional models [67]. With less feature engineering, deep neural networks can generalize better to unseen feature combinations through low-dimensional dense embeddings learned for the sparse features [15].

Most of the neural network applications applied in the question retrieval or question answering tasks, have the same goal: to learn distributed vector representations of documents, on which a similarity metric (usually the cosine distance) can effectively measure the matching degree [61, 75, 9, 25, 8, 67].

Training a neural network to become a ranking model, can be achieved by three methods, namely the pointwise , pairwise or listwise method. These, differ in how many documents one considers at a training step in the loss function calculation. The pointwise approach looks at one document at a time, finding the error in the similarity estimation of a document to the query. The pairwise approach looks at pairs of documents, comparing them with the ground truth ordering, and the listwise approach looks at a list of documents and compares it with the optimal ordering. In this work, neural network based models are used for the Question Retrieval task with the pairwise approach adopted.

3.2 Multi Label Classification

Multi-label classification has many real-world applications; among them, is the document organization into several not mutually exclusive categories and the automatic labeling of resources in the Web, such as texts, images, music, or video ([45]). Labeling Web resources with the objects that appear in an image, the genres derived by movie trailers or songs, constitutes a way to categorize and build taxonomies in Web pages where the number of new resources is enormous and the manual categorization by human intervention is infeasible.

Multi label classification should not be confused with multi-class classification. The former allows an object to belong in multiple classes with arbitrary probabilities for each class, while the later allows an object to belong only in one class (probabilities of belonging in each class sum up to one). Examples for multi-labeled data are songs characterized as both rock and ballad, patients diagnosed by multiple deceases at the same time [73], genes functionality [19], or images showing beaches, sunsets and animals at the same time (semantic scene classification) [63, 69].

The learning of multi-labeled data can be divided in two main approaches: data transformation and algorithm adaptation. The idea of data transformation, is to modify the data representation and create one of the following frameworks:

- One versus Rest: q binary classifiers must be trained, where the labels are assumed independent (label correlations are not modeled). [78]
- Multi-Class: normal classification framework with 2^q classes (all possible label chains are converted into different labels).
- All versus All: $\frac{q(q-1)}{2}$ binary classifiers must be trained.

Then, the output of each method needs reconstruction in order to become a set of predicted labels. The idea of algorithm transformation is to adapt an algorithm. Examples include:

- kNN adapted to assign training example x to the most common labels of the k nearest neighbor. [36]
- Neural Networks (NNs) with multiple outputs, i.e. the output is in the form of $[num\_of\_samples, num\_of\_classes, 2]$ and softmax activation followed by cross entropy loss is utilized. [41]
- Adaboost minimizing hamming loss or ranking error. [64]

The main difficulty is to build a model capable to predict several outputs at once. Some approaches are easily adaptable (like NNs), whereas others require more effort. Most of the existing approaches assume labels are independent for simplicity. However, in many scenarios this is not the case; for instance in the AskUbuntu forum the probability of the label “dual-boot” would be higher if the label “installation” is also relevant (see fig. 6). Recent work [16, 81, 2] shows potential improvement when incorporating the labels’ dependencies.

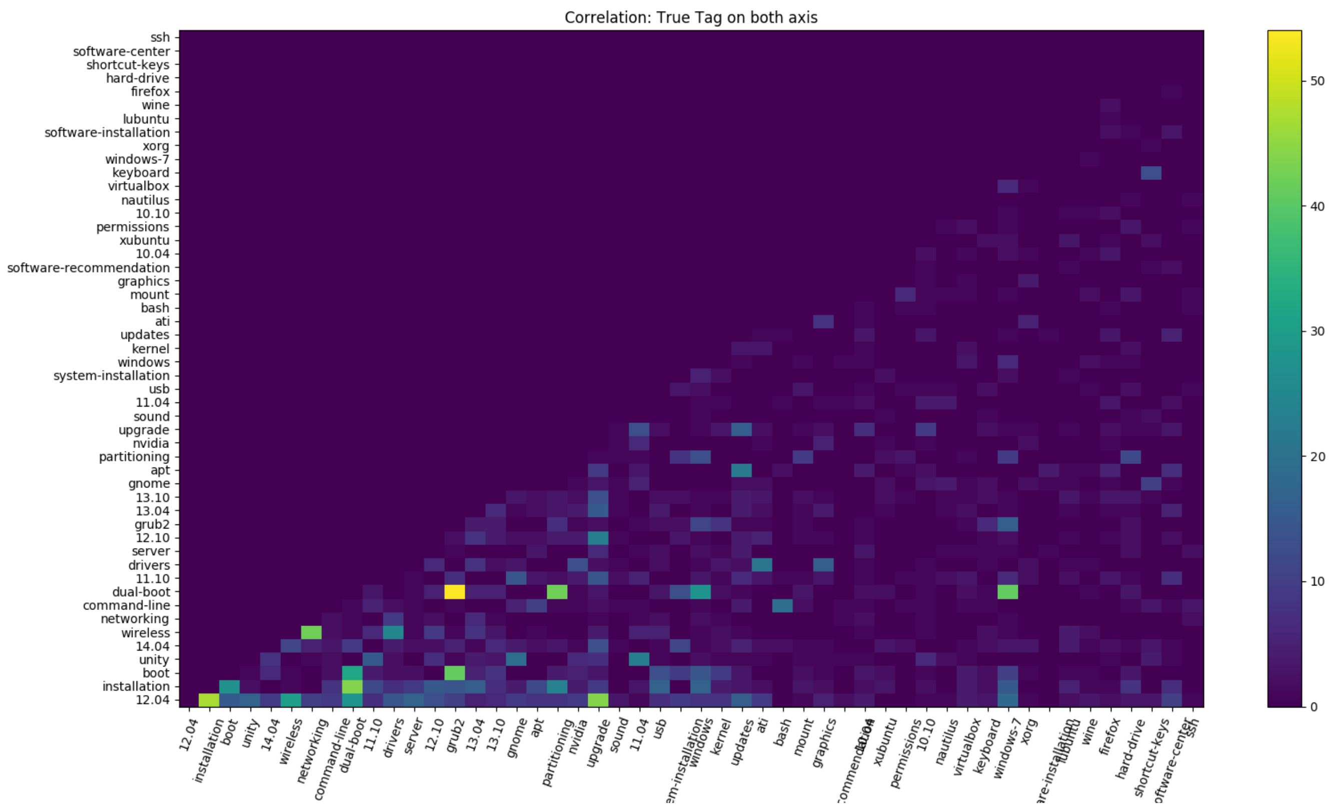


Fig. 6. Correlation matrix of the top 50 tags.

A naturally arising problem in multi labeled datasets, is the high dimensionality. In a scenario with thousands of labels, it is expected that some of the labels appear a lot rarer than others. This can be visualized in the domain of AskUbuntu tags in fig. 12, where the distribution of labels is not uniform; instead, it is very much skewed. Additionally, the transformations applied to multi-labeled data tend to increase the imbalance between labels. This phenomenon appears in the case of Binary Relevance (One vs. Rest framework), where each label's model takes as positive only the instances in which that certain label appears and the rest of the samples as negative, leaving a skewed distribution for each label. The phenomenon is even worse if the label is already a minority label ([34]).

Classifiers like NNs, that can output a distribution of scores or probabilities for all the labels are called distribution classifiers. Simple heuristics can be applied to them in order to output a set of labels; for example, the selection of labels for which the score is higher than a threshold, or greater than a percentage of the highest score [73], in other words, transforming the problem to output a list of non-ranked labels [10]. Additionally, the problem can be casted into a mere ranking problem. In [22], SVM is trained with a ranking loss (the average fraction of pairs of labels that are ordered incorrectly) and learns to rank the labels.

In this work, multi label classification is used for tag recommendation. The Binary Relevance approach is utilized training Logistic Regression models for each tag, as well as the Neural Network approach with multiple outputs giving the probability of each tag.

3.3 Tags Recommendation

Tags have become widely popular on web applications, supporting easy and fast posts' description, and this has led to an increase in the tag recommendation literature, aiming to increase the quality of generated tags, in order to improve IR systems like content recommendation [4, 29].

Tags form folksonomies –categories based on freely chosen names– on the web. They specify an easy way to retrieve content on the web, like on CQA or image platforms or in Client Support systems. In any case, users are allowed to specify tags for resources (questions, images, client issues etc.), and this facilitates the organizing and sharing of the resources among the users. A customer support employee is supposed to search the company's database and find problems similar to the new ones, in order to re-use solutions. Similarly, a forum user is exploring historical questions by querying the platform with keywords he assumes to be helpful. It is in the user's responsibilities to guess what are the good terms to look up when searching for similar objects, therefore, the more experienced he is on the specific topic, and the domain in general, the better search he can do. If a user is new, he might lack knowledge and misunderstand the topic or its candidate solutions, thus the possibility to look up with the wrong keywords is high. Note that tags recommendation is different from keyword extraction algorithms. Keywords are words that must appear in the original question, whereas tags can be absent [78]. Tag recommendation makes the question posting easier for the users, as it recommends good tags at the time the user types his query. Correct tags are very useful to new users, that are not familiar yet with the domain, and avoids the wrong expansion of the tag-based object taxonomy. As a result, it can also improve the object retrieval by ensuring consistency among the tag usage.

Existing work on tag recommendation can be split into the object-centered approaches and the personalized (user-centered) ones [70]. However, user-centered approaches might not be efficient, since only few users perform tagging extensively [24]. The document-centered approaches are more robust because of the rich information contained in the documents.

Tag recommendation on social sites is harder than other, structured domains (like scientific documents) with controlled vocabularies [70]. The heterogeneous nature of web pages consist both varied length of the html pages, and unpredictable tag distribution. Tags might include the same compound word as different tags; either separated by spaces, hyphenated, or joined. Previous work on Software Forums has pre-processed the dataset by cleaning the labeled tags [36, 3]. Three systems, TagCombine [79, 78] and EnTagRec [76], were proposed as state-of-the-art systems for tag recommendation in information sites including the AskUbuntu forum.

EnTagRec is a method composed of bayesian and frequentist inference. The bayesian inference component –inspired by the LDA [7]– models a question as a probability distribution of tags, and a tag as a probability distribution of words –that appear in the questions tagged by it. The frequentist inference component considers a question as a set of tags attached to it, and a tag as a set of words appearing in questions tagged by it. The two components are linearly combined to produce the resulting predictions. A recently revised approach, EnTagRec+ [77], leverages user information as well, along with initial tag sets that the user might provide. TagCombine method is explained in section 5.4.1 and we use it as our baseline tag recommendation system.

Tags, created either individually or collaboratively, can serve users in CQA [3]. They can be utilized as a source of meta description, which can be further used for user or domain modeling, personalization or recommendation. In this research, tags are utilized to model the topic connections between similar/dissimilar questions and boost the question retrieval model's performance.

Existing tag recommendation methods are either graph-based [57] or content-based(e.g. [46]). In this work, similarly to our question ranking model, we model tag recommendation as a content-based approach.

3.4 Multi Task Learning

Multi-task learning (MTL) has the potential and the idea to create robust models trained on limited data. Recently it gained a lot of attention in the NLP area with the scientists trying to guess what tasks can be trained together to make robust models.

The mechanisms of multi task learning [13] are given below:

- Statistical Data Amplification: two different tasks having different noise patterns, can share the feature layer and learn better representations by averaging their noise patterns.
- Representation Bias: a multi task learning model prefers hidden representations that other tasks prefer, in other words, the optimization ends at a local optimum common for all tasks.
- Eavesdropping: if a network finds it hard to model task A, but it can easily model task B, then, we can jointly train it on both tasks, and the knowledge from task B can be transfered for modeling task A; and vice versa. This results in a communication channel.
- Attribute Selection: when a network is trained on more than one task, it learns to identify better which features are useful for each task.

Summarizing, the focus on a single task, can lead to the ignorance of important information that might be helpful on the main task, whereas sharing representations between related tasks, can result to more generalized models, that perform better on the main task. They do this, by preferring sparser solutions which explain multiple tasks. [59]

The problem of MTL is mainly split in two questions. The first one is about the model architectures, and concerns the question: which layers should be shared among different models of different tasks? The second is about the auxiliary tasks, and concerns the question: which tasks and what objectives can boost the separate models' performances? Both questions, require trial-and-error –in other words, parameter tuning– which is easier achieved if domain knowledge exists, and the scientist tries to guess how the tasks must be incorporated and co-operate within one framework. An easy and frequently used approach in MTL is to create a model that shares all layers' parameters except the last one's - which differs for each of the tasks-to-be-learned [14]. Another architecture, is to use different parameters for each task, with restrictions on the parameters' distance between the tasks.

The most common approach for defining an objective function within a joint learning framework is to make a linear combination of the individual tasks' objectives. However, a recent work [40] shows that performance of multi task systems depends strongly on the relative weighting between each task's loss and the authors propose a method to automatically find good weights based on the uncertainty of each task. Additionally, all the tasks in total, or all the tasks after clustered by their similarity, share a common regularization parameter; this ensures that the training of a simple task takes into consideration what is learned for other tasks. Furthermore, the training of multiple tasks when some of the tasks are more related than others, has been proved to be better implemented in a sequential manner, with the easier tasks followed by the harder or conditional ones. However, the most common method, is to simultaneously learn all the tasks.

We should note that MTL is strongly related to Transfer Learning. The former aims to perform well on more than one tasks, while the later aims to boost the performance of one targeted task [53]. Fine tuning is a basic example of multi task learning (or transfer learning), where different learning tasks are leveraged by considered as pre-training steps. A simple, yet famous example is the pre-training of the word-embeddings used to structure a language model or a text-encoder in general, on the text corpora of interest, i.e. corpora related to the final task. This approach is widely used and proved to be effective [54, 56]. Another successful MTL application is the simultaneous training of a machine translation model to learn translations from language A to language B and vice-versa [32].

Most of the work in MTL considers more than 2 tasks and their main concern is how to couple them, according to their relatedness (example work in [38]). This happens to avoid negative transfer, that is, in case of tasks not closely related, the shared information can be unsuccessful, and even hurt the performance.

In this work, we use multi task learning both to pretrain the word embeddings and to transfer knowledge from the tag recommendation task to the question retrieval one.

4 Problem Definition and Methodology

The main problem addressed in this work is Question Retrieval. The definition of the problem follows the notation in table 3 which remains constant to the rest of the paper.

Notation	Explanation
q	a query question
D_q	a set of candidate questions to be ranked in respect to q
D_q^+	a set of questions similar to q
D_q^-	a set of questions dissimilar to q
V_q	encoded representation of q
d	a candidate question
d^-	a negative question (in regards to the query referred in the context)
d^+	a positive question (in regards to the query referred in the context)
$\|q\|$	the length of q in words
w_q^i	the word in the i-th position of q
v_q^i	the word embedding of the word w_q^i
V	vocabulary of words
W	word embeddings matrix
N_s^-	number of negative samples per query question
N_q^+	number of positive candidates for question q
ST	set of tags
$Tags_q$	set of tags assigned to question q
$tags_q$	sparse boolean representation of $Tags_q$
θ	network parameters

Table 3. Notation used in the work.

We have a set of query questions, $Q = \{q_1, q_2, q_3, ..., q_{|Q|}\}$, for which a search engine has returned L candidate questions for each query, $D_q = \{d_q^1, d_q^2, d_q^3, ..., d_q^{|D_q|}\}$. Considering binary annotations, some of candidates are similar to the query and others are not. A pair of questions (q_i, d_j) is a positive pair if d_j is similar to q_i, otherwise it is a negative pair. Accordingly, the candidate question, d_j, is said as a positive or negative question, correspondingly. A query is shown to the system along with the list of candidate questions, and the system is responsible for giving a ranking, with the most similar candidate being on top. Following is an example:

Query q_4 with the candidate list $[d_2, d_5, ..., d_{12}]$ are passed through a golden ranking system which outputs the ranked list $[d_3, d_2, ..., d_{12}]$. This means that $S(q_4, d_3) > S(q_4, d_2) > ... > S(q_4, d_{12})$, based on S, a function that gives the real similarity degree between questions.

The methodology followed in this work to build our basic neural network-based retrieval model is given in section 4.1. Then, in section 4.2 we show the approach utilized for building the neural network-based model that address the tags recommendation problem. This is necessary to address our research question RQ1, and is a prerequisite for addressing the the research questions, RQ2 and RQ3. To help the reader navigate through the methodology, we now give the definition of our auxiliary task, tags recommendation.

We have a list of query questions, $Q = \{q_1, q_2, q_3, ..., q_{|Q|}\}$, each one annotated with a set of tags, i.e. $Tags_{q_i} = \{tag_1, tag_2, .., tag_{|T_{q_i}|}\}$, where $tag_i \in TH$, and TH is the set of all tags used in the corpus. A query question is shown to the system, and the system is responsible for giving a ranking of all tags in TH, with the most probably matching tag being on top. Now the top T tags can be retrieved and recommended as the tags matching to the query.

Finally, in section 4.3 we show the methods used to address the research questions RQ2 and RQ3, precisely, how we use pretraining and multi task learning to transfer knowledge from the tags recommendation task to our main task: question retrieval.

4.1 Question Retrieval Model

To build a neural ranking model for question recommendation based on the questions' texts, we build a neural encoder which is responsible to transform the texts into high-dimensional embeddings. Various encoders are tested based on recurrent or convolutional operations, precisely, either convolutional filters, LSTM units, GRU units, bidirectional LSTM units or bidirectional GRU units. The hidden multidimensional space ideally learns representations of questions whose features model the question similarity. In other words, similar questions lie in the same clusters in the hidden space. Therefore, when the questions' embeddings are obtained, the cosine similarity is utilized to score and rank the candidates. Following there is a more detailed explanation.

4.1.1 From Input Representation to the Encoded Representation

A query question, q, is represented by title t, and body b, as the sequences of words $[w_t^1, w_t^2, ..., w_t^{|t|}] and [w_b^1, w_b^1,$ $..., w_b^{|b|}]$, respectively. Word embeddings (W) are the distributional vectors $v \in R^{dx|V|}$, where V is the vocabulary of all words. For convenience and ease of lookup operations in W, words are mapped to integer indices $1, ..., |V|$. Now the input sequences t and b can be transformed into the sequence matrices $t_{mat}, b_{mat} \in R^{|t|xd}$.

$$t_{mat} = \begin{bmatrix} - & v_t^1 & - \\ - & v_t^2 & - \\ - & ... & - \\ - & v_t^{|t|} & - \end{bmatrix} b_{mat} = \begin{bmatrix} - & v_b^1 & - \\ - & v_b^2 & - \\ - & ... & - \\ - & v_b^{|b|} & - \end{bmatrix}$$

The sequence matrices of title and body are passed through the *Encoder* which outputs two embedded representations, and we average them to get our final question representation, V_q. The Encoder is consistent of the following. An Input Layer (the word embedding matrix), two Dropout Layers, Dr with dropout probability p and an Encoding Layer (CNN, LSTM, GRU, Bi-LSTM or Bi-GRU), H. The process is visualized in figure 7 and equations 11- 14.

$$t'_{mat} = Dr(t_{mat}, p) \qquad b'_{mat} = Dr(b_{mat}, p) \tag{11}$$

$$h_t = H(t'_{mat}) \qquad h_b = H(b'_{mat}) \tag{12}$$

$$h'_t = Dr(h_t, p) \qquad h'_b = Dr(h_b, p) \tag{13}$$

$$V_q = (h'_t + h'_b) * 0.5 \tag{14}$$

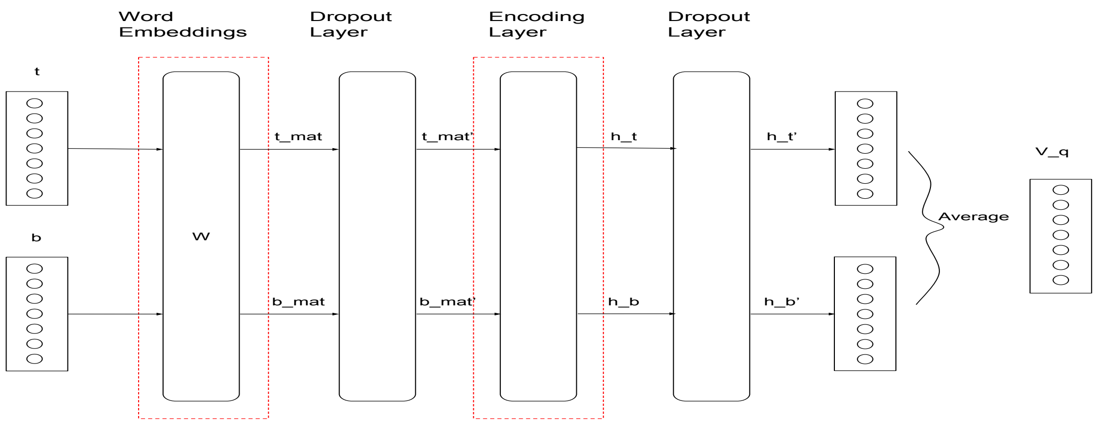


Fig. 7. Question Retrieval Model.

4.1.2 Training the Encoder

In Fent et al. 2015 [25] the authors have experimented with various architectures for text similarity on answer selection and the one used in this work (fig. 8) –same encoder utilized for both query and candidate representations and cosine similarity used for scoring– is one of the two architectures they report the highest performance.

The network is fed with a query question, q, N_q^+ positive candidate questions, $D_q^+ = \{d_1^+, d_2^+, ..., d_{N_q^+}^+\}$ and N_s^- negative candidate questions, $D_q^- = \{d_1^-, d_2^-, ..., d_{N_s^-}^-\}$, which we denote as a query tuple. It then produces the representations $V_q, V_{d_1^+}, ..., V_{d_{N_q^+}^+}, V_{d_1^-}, ..., V_{d_{N_s^-}^-}$, respectively, and the matching scores $S(V_q, V_{d_1^+}), ...,$ $S(V_q, V_{d_{N_q^+}^+}), S(V_q, V_{d_1^-}), ..., S(V_q, V_{d_{N_s^-}^-})$ are afterwards calculated using the cosine similarity.

The cosine similarity for two vectors, $\vec{a}$ and $\vec{b}$, is defined as: $cos\theta = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||}$. This gives the cosine of the angle between the directions of the normalized vectors. In other words, no matter how far in the hidden space the two questions fall due to the numeric differences of their features, they can be closely related if they focus at the same features. The resulting domain is $[-1, 1]$, with -1 denoting two opposite vectors, 0 denoting two vectors very different with no correlation, and 1 denoting two very related vectors.

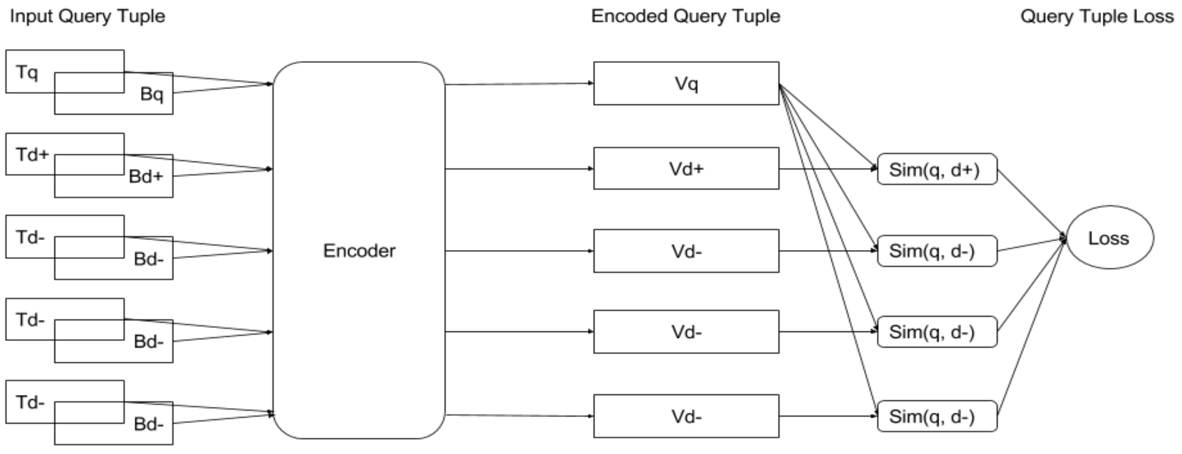


Fig. 8. Question Retrieval Model framework.

Now we can rank the candidates from the most matching one to the least matching one, and define a loss function, L, to realize the learning process. We use the max-margin (hidge loss), used in previous work [44, 75, 25].

The max margin loss, L_{q,d^+,d^-}, can be explained as the error of estimating similarity $S(V_q, V_{d_x^-})$ higher than $S(V_q, V_{d^+}) - \epsilon$, where ϵ, is a small positive (the margin), d_x^- is a negative candidate and d^+ is a positive candidate. $L_{q,d^+,d^-} = \max(S(V_q, V_{d^-}) + \epsilon - S(V_q, V_d^+), 0)$.

If this condition is satisfied we imply that the model either ranks the positive question below the negative one, or it ranks the positive question above the negative one with an insufficient margin; in either case, the hidden space fails to model the data characteristics and their correlations, and its parameters, θ are updated towards the inverse of $\frac{dL}{d\theta}$; otherwise the loss is zero and parameters are not updated. Notice that L_{q,d^+,d^-}, unlike the initial hinge loss used for SVMs –which simply assigns a constant penalty of 1 to every pair of questions wrongly ranked– takes into account the degree of difference between the positive and negative scores. This is especially important in case where exact relevance degrees are given instead of boolean relevance values.

Since multiple negative candidates are coupled with a positive one, we experimented with three variations of the max-margin loss for a query, used in other work [1, 21].

The first loss (equation 15) we test, considers all the pairs of similar and dissimilar questions, defining a relaxed loss function. The second loss, defined by equation 16 considers only the negative example for which the ranking loss is higher, and compares its score with all scores obtained by the positive candidates. The last loss function we test, defined in equation 17, considers only the pair of (positive, negative) candidates with the higher ranking error. To utilize batch training, we define the batch loss in equation 18.

A visualization of the training process is shown in figure 8.

$$L_q^{QR1} = \frac{1}{|D_q^+| \cdot |D_q^-|} \sum_{d^+=1}^{|D_q^+|} \sum_{d^-=1}^{|D_q^-|} \{L_{q,d^+,d^-}\} \tag{15}$$

$$L_q^{QR2} = \max_{d^- \in D_q^-} \frac{1}{|D_q^+|} \sum_{d^+=1}^{|D_q^+|} \{L_{q,d^+,d^-}\} \tag{16}$$

$$L_q^{QR3} = \max_{d^- \in D_q^-, d^+ \in D_q^+} \{L_{q,d^+,d^-}\} \tag{17}$$

$$L^{QR} = \frac{1}{|B|} \sum_{q \in B} L_q^{QR} \quad , \qquad L_q^{QR} \in \{L_q^{QR1}, L_q^{QR2}, L_q^{QR3}\} \tag{18}$$

4.1.3 Inference

The network is fed with a query question, q, and N_c candidate questions, $D_q = \{d_1, d_2, ..., d_{N_c}\}$. A pass through the network produces the representations $V_q, V_{d_1}, ..., V_{d_{N_c}}$, respectively. The similarity scores $S(V_q, V_{d_1})$, $...S(V_q, V_{d_{N_c}})$ are calculated, and the model outputs a ranked list of the candidates.

4.2 Tags Recommendation Model

We model the tags recommendation problem as a multi label classification problem, implemented with a neural network.

First, we transform the target set, $Tags_q$, of tags assigned to question q, into the sparse boolean representation $tags_q \in R^{|TH|}$, where

$$\forall i, tag_i \in TH, tags_{q_i} = \begin{cases} 1, & \text{if } tag_i \in Tags_q \\ 0, & \text{otherwise} \end{cases}$$

Next, we build a neural encoder, in the same way explained in section 4.1.1, to transform a question from its title and body word sequences to the multi-dimensional representation V_q. The output of the encoder, is passed to a simple feed forward network, which is responsible to output the probabilities of each tag as being a good match for the given question.

To realize the learning process, the sigmoid binary cross entropy loss function is used [42, 52, 27]. For one training example the loss is defined by equation 19. To utilize batch training, we define the batch loss in equation 20

$$\begin{aligned} L_q^{TR} &= -log(\sigma(logits_q)) * tags_q - log(1 - \sigma(logits_q)) * (1 - tags_q) \\ &= \sum_{i=1}^{|TH|} -log(\sigma(logits_{q_i})) * tags_{q_i} - log(1 - \sigma(logits_{q_i})) * (1 - tags_{q_i}) \end{aligned} \tag{19}$$

$$L^{TR} = \frac{1}{|B|} \sum_{q \in B} L_q^{TR} \tag{20}$$

where B is the mini-batch of questions and $logits_q$ is the neural network output vector of the last layer, which defines the scores of tags as being relative to the question and are found by the equations 21, 22 and 24 in the case of a 1 Layer MLP and equations 23 and 24 in case of an SLP, as visualized in figure 9.

$$\vec{h} = relu(\vec{W_h} \cdot \vec{V_q} + \vec{b_h}) \tag{21}$$

$$\vec{o} = \vec{W_o} \cdot \vec{h} + \vec{b_o} \tag{22}$$

$$\vec{o} = \vec{W_o} \cdot \vec{V_q} + \vec{b_o} \tag{23}$$

$$logits_q = sigmoid(\vec{o}) \tag{24}$$

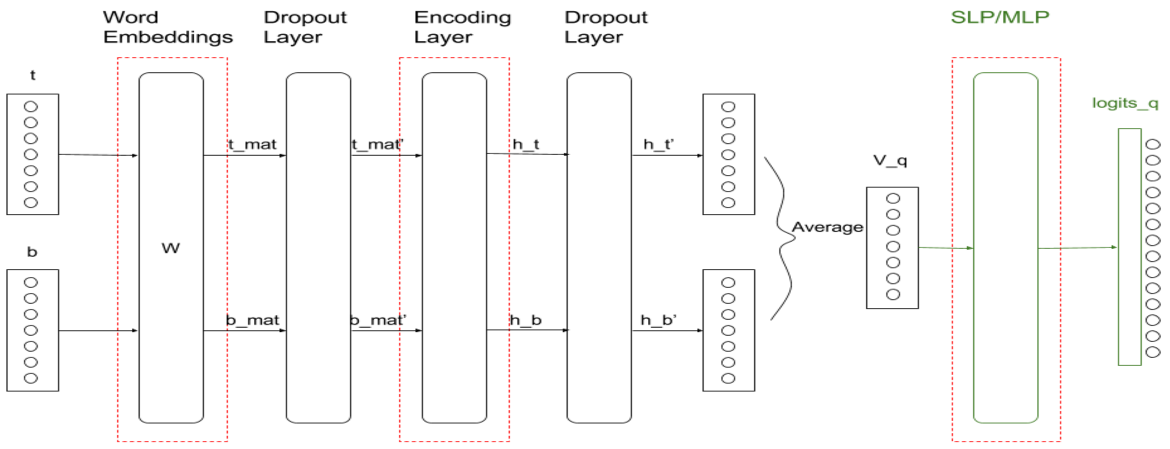


Fig. 9. Tags Recommendation model architecture. The red boxes indicate the network parameters.

4.3 Embedding the Tags Recommendation task within the Question Retrieval problem

This section defines how the category information can help improve the question retrieval neural network based ranker. A first approach is via pretraining the neural ranker on the tags prediction task, and a second approach is via jointly training the network on the two tasks.

4.3.1 Question Retrieval model pretrained on Tags Recommendation task

Figures 7 and 9 visualize the network architecture of each task. Weights of the Question Retrieval model are also weights in the tags recommendation one. A straightforward way to enhance the neural based ranker, is by training the tags recommendation model and then fine tuning the parameters on the target task. With fine tuning we mean the driving of the initialized (good) weights to a local optimal point of our main task. This approach is examined in two ways:

- Learning all the trainable parameters during the tags prediction training, and then initializing the QR model Input and Encoding layers.
- Learning the encoder and output layers during the tags prediction training, while using the word embeddings trained on the language model, and then initializing the QR model encoder layer, while using the same constant word embeddings.

4.3.2 Joint Training of Question Retrieval and Tags Recommendation tasks

The goal of this method is to use the auxiliary representations of the TR model to help the main task (QR) achieve better performance. Before explaining our joint learning implementation, a new problem definition must be given for our main task (QR). The new task is an expert ranking system, which receives not only the questions' title and body, but the questions' tags as well.

Given a query question, q, and its assigned tags, $Tags_q$, along with candidate questions, D_q, and their assigned tags, $Tags_d \forall d \in D_q$, rank the candidate questions with the most similar to the query being on top.

Both the models of out main task (QR) and auxiliary task (TR) learn from the same features, therefore the tasks are assumed to be closely related and we believe that a simple multi-learning approach where only the Word Embeddings and Encoding Layer are shared between the tasks can help achieve our goal [13].

Consequently, the architecture of the model for the new task is the one depicted in figure 9, with the *Encoder* being shared across the tasks and visualized in fig. 7. To accomplish the simultaneous learning of the tasks, an overall loss function must be defined, and we use a linear combination of the individual tasks' losses with a common regularization factor 25. Considering both losses and a generalization factor that accounts for both models parameters helps the network to predict well in both tasks, and each task to help each other during the learning. We visualize the process in figure 10. The weighting factors α and β are found empirically, $c = 1e^{-}5$.

$$L = \alpha L^{QR} + \beta L^{TR} + C \sum_{\theta_i \in \theta} ||\theta_i|| \tag{25}$$

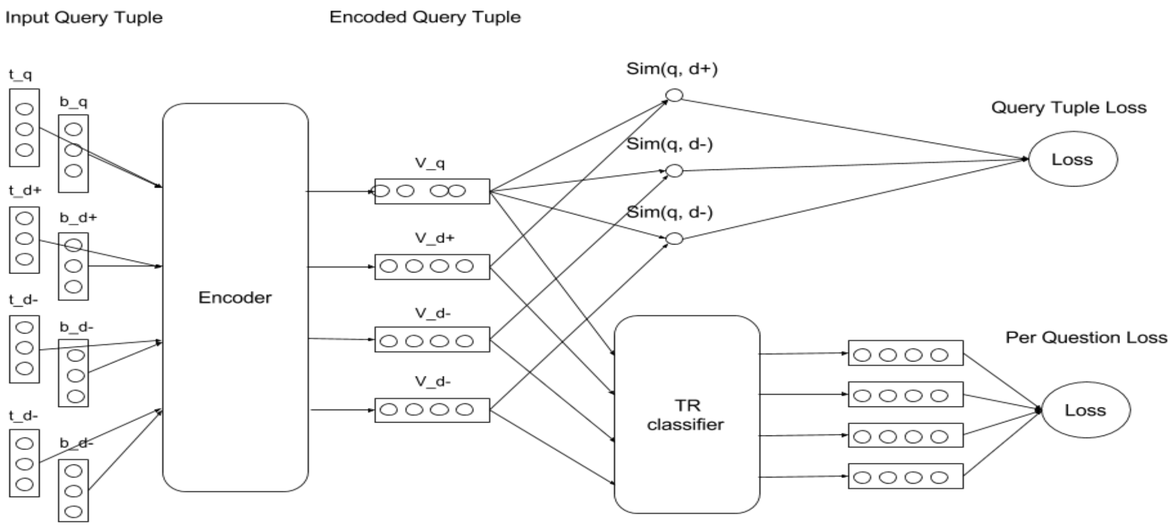
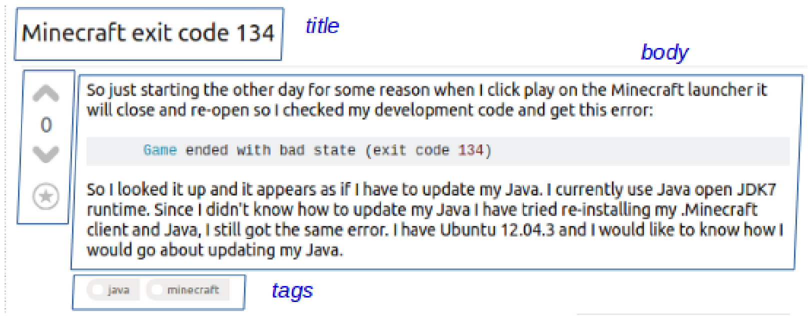


Fig. 10. Multi Task Framework.

5 Experimental Setup

5.1 Data

In this research, data from the Ask Ubuntu forum (sub-forum of the Stack Exchange Web platform) are used. Precisely, the data is retrieved by Dos Santos et al. 2015 [62], as a May 2014 dump, and is used in other work[62, 44, 28, 8]. This dataset consists of 167765 unique questions. Only a very small amount of questions consisted of similar annotated questions, therefore domain experts annotated part of the data to make proper evaluation sets. An example of a question posted in the forum is illustrated in fig. 11. It should be noted that datasets for question similarity often face the problem of incomplete annotations, since extraordinary manual work is required to annotate every pair of questions. This is also the case for other publicly available data (SemEval 2017 task 3 subtask E data [4] and Quora Duplicates Dataset [5]).



Minecraft exit code 134

So just starting the other day for some reason when I click play on the Minecraft launcher it will close and re-open so I checked my development code and get this error:

```
Game ended with bad state (exit code 134)
```

So I looked it up and it appears as if I have to update my Java. I currently use Java open JDK7 runtime. Since I didn't know how to update my Java I have tried re-installing my .Minecraft client and Java, I still got the same error. I have Ubuntu 12.04.3 and I would like to know how I would go about updating my Java.

Fig. 11. Example of question from AskUbuntu forum.

The text of questions is kept the same as used in [62, 44, 28, 8] for comparison reasons and for simplicity. Questions asked in the forum are concerned with technical issues and contain mostly software ontology, URLs, computer locations and other hard to process types of text. Some of the possible pre-processing steps are the identification and replacement, removal or partial-removal of emails, URLs, and file locations. Such text however, can contain important part of the questions and their similarity specification. In the published data, a language model [50] was trained on questions from the StackExchange forum and Wikipedia and word embeddings were obtained for a vocabulary that contains all the important technical words. The data were tokenized by the Stanford Tokenizer **??** and only parts inside the “code” tags - i.e. unique commands and programming codes- were removed.

The data is split into a training set, a development and a test set. Both development and test sets contain 200 query questions, each with 20 candidates, resulting to 40055 and 3735 unique questions appearing in the development and test set, correspondingly.

5.1.1 Data Sets for Question Retrieval

The training set for the question retrieval model is created by pairing each query with its positive (less than ten) queries and twenty negative candidates sampled from the entire corpus, which are considered good similarity candidates (informative examples), in order to have a significant impact on the effectiveness of the model [62]. The development and test sets contain tuples of queries, q, paired with twenty candidate questions, $d_1, d_2, ..., d_{20}$. Some questions appearing in the evaluation files (either as similar or candidate questions) appeared in the training file as well; either as queries, similar questions, or candidate questions. We note that the challenge for question retrieval is to learn to compare various textual questions, and the comparison of questions A and B while training, is different from the comparison of questions C and B who appear in the test phase. However, if the network is trained with various tuples containing B, it can be more familiar with B when it will encounter it in a test phase. Thus, for farer evaluation, any question appearing in the evaluation phase is removed from the training set. This is in agreement with the given datasets of the SemEval 2016-17 task 3 competition. It can be argued that performance will be low if the questions appearing in the test set talk about very rare topics that are not encountered at all while training. However, an investigation of the data ensured that similar topics appear for every test query, therefore we removed those queries from the training.

[4]http://alt.qcri.org/semeval2017/task3/index.php?id=data-and-tools
[5]https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

5.1.2 Data Sets for Tags Recommendation

The training set for the tags recommendation model is defined as all the questions appearing in the question retrieval training set. One question text is given as the input, and the question tags are given in the form of a boolean vector, as the corresponding output.

The tag set used, TH, is a subset of all the tags appearing in the corpus, selected to satisfy the rule $50 <= tag\_occurrences <= 10000$. If a tag appears in the site very rarely, it happens either because it is a bad choice –e.g. a misspelling–, or because it is a rare topic; in any case the tag is inappropriate for others to recognize it [85]. As shown in fig. 12 the tag frequency drops dramatically, with the top 100 tags constituting the 82.3% of all the tags and the top 1000 tags constituting the 98.8% of all tags. However, the removal of some tags leaves some of the questions with zero labels. To be more precise, 3961, 88 and 67 questions are left with zero tags in the training, development and test set, respectively. Since the model is trained with binary cross entropy, these questions can still be used during the training, with the network logits potentially resulting to zero for every class. During the evaluation phase however, they are ignored, since the absence of labels results to zero precision and undefined recall. Note that these data are silver true annotations rather than golden true annotations, since questions might be addressing more than their assigned tags.

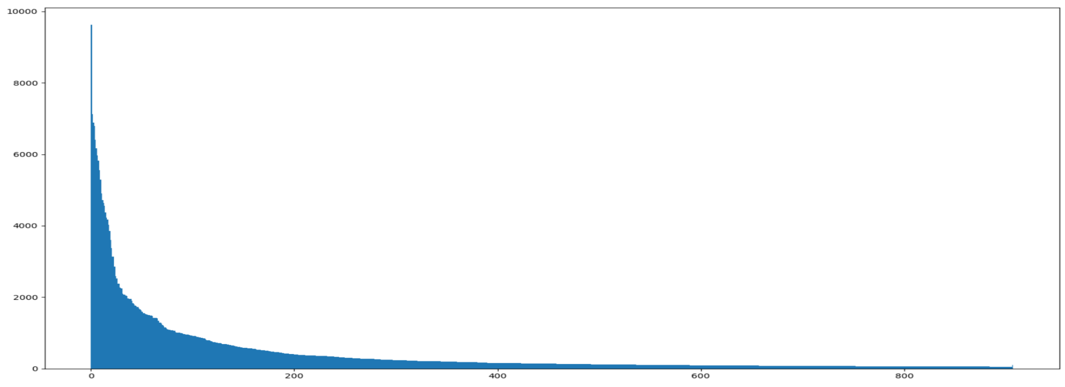


Fig. 12. Tags frequency distribution on selected tags with $50 <= occurrences <= 10000$

5.2 Pre-trained word embeddings

As already mentioned in section 5.1, word embeddings for this domain were obtained by training a SkipGram language model (with the Word2Vec tool). The trained embeddings were used as initial values in both the question ranking and tags recommendation models. These weights are then kept constant or updated. We keep the word embeddings of all experimented configurations as 200 dimensional vectors.

5.3 Implementation Details

- The weights (and biases) in the convolutional layers are initialized by the distribution $U(-0.5, 0.5)$. The same applies for the weights initialization of the last layer (MLP). The weights of the LSTM and GRU units are initialized with the default parameters of Tensorflow, which is zero.
- The activation functions play an important role in the neural networks, since they transform the linear functions into non-linear ones, making the network capable for non-linearly separable problems. ReLU is currently the most commonly used activation function, but it can easily fall into gradient problems. $Sigmoid$ and $\tanh$ are similar, and both constrained into the bound $(-1, 1)$, with the gradients of $\tanh$ being steeper. In this work, $\tanh$ activation function is used across the networks for simplicity.
- Dropout is applied at the input and output of the encoder; not in the hidden neurons.
- The bidirectional models (Bi-LSTM and Bi-GRU) do not share parameters between their forward and backward structures.
- The Adam optimizer is used in all the experiments, with a learning rate of 0.001.
- Only one convolutional stacked layer is used. Other works have shown that only when very big amount of data are used, the use of more than one stacked layers becomes useful, boosting the performance ([80]).

- Early stopping is utilized to avoid overfitting.
- The implementation [6] is written in Python utilizing the machine learning library Scikit Learn and the deep learning library Tensorflow.

5.4 Baselines

5.4.1 Question Retrieval Baselines

The baseline model we use to compare our neural network-based models in the task of Question Retrieval is the Okapi BM25 explained in section 3.1.

5.4.2 Tags Recommendation Baselines

Half the questions are assigned with tags that do not appear in their text. Additionally, the distribution of all the tags is very skewed (see fig. 12). These corpus' characteristics make the naive approaches of predicting the existing or the majority tag as inappropriate baselines.

Among the existing methods we choose to implement the TagCombine model which has simple implementation. We also test a binary relevance approach with Logistic Regression models which previously shown successful in multi label problems [48].

TagCombine
In [79, 78], a system called TagCombine is proposed based on three components: Multi-Label ranking, Similarity-Based ranking and Tag-Term Based ranking.

The first component, considers tag recommendation as a multi-label classification problem. A One-vs-Rest multinomial Naive Bayes model is trained on the BoW representation of text, modified to return the likelihoods for each tag, $p(tag_k/q) = \frac{p(tag_k)p(q/tag_k)}{p(q)}$, rather than 0/1 labels –in this way the evaluation as a ranking system arises naturally.

The second component, models historical tags on similar questions with frequencies, and recommends the most likely ones. More precisely, the most similar questions to the query question are retrieved based on the cosine similarities of tf-idf vectors. Then the probability of a tag $tag_k \in TH$ to be recommended is taken as

$$\frac{counts(tag_k)}{\sum_{tag_j \in TH, j \neq k} counts(tag_j)}.$$

The third component, models the probability of a tag tag_k to be recommended in a question $q = w_1 w_2 w_3 ... w_N$ as the product of the affinity scores (found by historical questions) of its terms and the tag,

$$p(tag_k) = 1 - \prod_{i=1}^{N}(1 - aff(tag_k, w_i)).$$

$$aff(tag_k, w_i) = \frac{counts(w_i, tag_k)}{counts(tag_k)}$$

With the three individual components, TagCombine uses a linear combination to give the final ranking. The linear weights are learned using a simple brute force on the parameter space $\alpha, \beta, \gamma \in [0, 1]$ to select the combination with the best results on a sample set.

Binary Relevance Logistic Regression
Given an input word sequence $q = w_1, w_2, ..., w_{|q|}$, the logistic regression model gives the tag-conditional probabilities via the linear logistic function. To model the target class tag_i, we use equation 26, where θ defines the model parameters. To estimate parameters, we minimize the loss function of all questions given in equation 27 (C is a free parameter).

$$P(tag_i = 1/q, \vec{\theta}) = \frac{exp(\vec{\theta} \cdot q)}{1 + exp(\vec{\theta} \cdot q)} \tag{26}$$

[6] https://github.com/christinazavou/deep_rank

$$\frac{1}{2}\vec{\theta}^T \cdot \vec{\theta} + C\sum_q log(exp(-\vec{\theta} \cdot q + c) + 1) \tag{27}$$

The probabilities of tags being assigned to the question, are taken as the scores for ranking. The only features used in this model are the tf-idf weighted uni-grams. In similar work it has been shown that the SVM performs well on multi-label problems [3], however due to its time complexity, which increases exponentially with data, it is not utilized here.

5.5 Evaluation Metrics

The problem of question retrieval as defined in section 4.1 composes a ranking problem, that is evaluated based on boolean labels. This means that no ranking is specified for each relevant question; instead, the only given is which candidate questions are similar and which not. Therefore, the goal of the ranking model is to rank all the relevant questions higher than the irrelevant ones.

Similarly, the problem of tags recommendation, as defined in section 4.2, composes a ranking problem, evaluated as well based on boolean labels. Consequently, we can define the same evaluation metrics for both tasks. To keep our results consistent and comparable with similar works [44, 79, 76, 78] we use the following metrics.

In order not to confuse the reader with questions and tags, we define the metrics around the information need of documents, like in generic Information Retrieval domains.

Precision at Rank k (P@k)
After cutting the retrieved list of documents to the top k results, $P@k$ is defined as the proportion of the k results that are relevant.

$$P@k = \frac{r}{k}$$

, where r is the number of r relevant documents retrieved up to rank k.

Recall at rank k (R@k)
After cutting the ranked list of documents to the top k results, $R@k$ is defined as the proportion of the k results that are relevant over the ideally retrieved k results.

$$R@k = \frac{r}{clip(R, 0, k)}$$

$$clip(R, 0, k) = \begin{cases} undefined, & \text{if } R \leq 0 \\ k, & \text{if } R \geq k \\ R, & \text{otherwise} \end{cases}$$

, where r is the number of r relevant documents retrieved up to rank k and R is the number of all relevant documents. Note that $R@k$ is undefined for queries with zero relevant documents, thus such cases are not evaluated.

Mean Average Precision (MAP)
MAP is the average precision averaged over all the information needs. Average precision is defined as

$$AP = \frac{\sum_r P@r}{R}$$

where r is the rank of each relevant document, R is the total number of relevant documents, and $P@r$ is the precision of the top-r retrieved documents. Therefore, MAP measures the quality of the precision across recall levels.

Mean Reciprocal Rank (MRR)
MRR is the reciprocal rank averaged over all the information needs. The reciprocal rank of a retrieved ranking is the multiplicative inverse (reciprocal) of the rank of the first relevant document.

$$RR = \frac{1}{r}$$

where r is the rank of the first relevant document retrieved. This measure evaluates best the case where the user is interested to see only one relevant document, and the best is to have it on top, therefore, the focus is on the ranking quality at the top of the retrieved list.

F1-score at rank k (F1@k)
Various competitors in the ECML PKDD Discovery Challenge 2009 evaluate their tags recommendation system with F1-score, which is the harmonic mean between P@k and R@k.

$$F1@k = 2\frac{P@k \cdot R@k}{P@k + R@k}$$

5.5.1 Statistical Significance

Statistical analysis is needed to be applied when IR experiments are made, in order to conclude significant results about the proposed systems. It is common to compare a baseline system to the proposed one, using paired test of significance, like the t-test. Statistical testing is responsible to tell whether the obtained increment between the systems' performance is due to randomness (null hypothesis), or whether the new system is indeed a better candidate (alternative hypothesis). Given a significance level, α, the test calculates a probability that performance difference is obtained by chance, $p - value$. If this $p - value$ is low enough (usually below 0.05), then the null hypothesis is rejected.

The t-test is applied with the assumption that the two experiments are made on two independent sets of equal samples. However, in Information Retrieval, this is barely ever the case; what is actually tested, are numerous systems running over the same set of queries, to get a fare comparison of which one performs better across the given data. Apart from this, it is very common to change some parameters in a model, and result to multiple systems, aiming to compare many of them against each other. If m systems are going to be compared the one against the other, then $\frac{m(m-1)}{2}$ paired tests need to be taken. This does not only become frustrating, but it also gives rise to the Multiple Comparisons problem [12].

Multiple Comparisons Problem

To make fair comparisons of information retrieval systems, statistical significance tests are usually utilized to drive us in the right conclusions. With a significance level α, a statistical paired test can give the wrong conclusion with probability $1 - \alpha$. If k paired tests are made, this probability –of getting at least one wrong result out of the k tests– increases to $1 - (1 - \alpha)^k$. This probability is known as the familywise error rate and is induced by the Multiple Comparisons Problem.

The Multiple Comparisons Problem is fortunately getting more attention in the research community these days, aiming to reduce the wrong results being published. Some approaches have been proposed on addressing this problem. Among the most famous is the Bonferroni correction [68], which simply reduces the critical $p - value$ for an individual test, to $k * p - value$, and its variation, the Benjamini-Hochberg correction [6].

In this thesis, the randomized Tukey's Honest Significance Difference (HSD) is applied, which is found as the one of the most preferable and powerful techniques. The randomized Tukey's HSD test takes the entire set of systems and simultaneously decides whether each pair has significant difference or not, while controlling a global statistical significance.

The Tukey's method tests at first the omnibus hypothesis, $M_0 = M_1 = M2 = ... = M_N$, where M_i is the performance of system i. If this hypothesis can not be rejected, it means that all the systems have the same performance, and there is no need for further tests. Otherwise, the Tukey's null hypothesis is tested, which assumes that all samples are taken from the same distribution. The test calculates a range value (the honest significance difference) as the maximum difference between a pair of system performance means. If a difference between two systems' means exceeds this range, then there is significant difference among those systems.

The test is implemented following the algorithm given in Sakai 2013 [60].

6 Results and Discussion

6.1 Neural Network-based Question Retrieval Model

For each type of network, we make multiple experiments. Statistical significance of their comparisons is shown in tables 18 and 19 in Appendix B.

We first see that the loss functions L_q^{QR2} and L_q^{QR3} perform similarly and give satisfactory performances 4. There is no statistically significant difference between these two functions. On the other hand, the loss function L_q^{QR1} gives very degraded results 4. We hypothesize that taking all the pairs of positive-negative questions gives a form of light penalization and the model ignores bad rankings in a list where good rankings are also obtained. This loss is probably only successful when millions of annotated data are used –like in [21]. The result that loss functions L_q^{QR2} and L_q^{QR3} are superior to L_q^{QR1} is compatible with the argument that loss functions in IR should are better defined at the level of queries rather than individual questions or question pairs [55].

	Loss	Test MAP	Test MRR	Test P@1	Test P@5
LSTM	L_q^{QR1}	48.47	61.70	46.77	38.71
	L_q^{QR2}	57.73	70.24	53.75	44.19
	L_q^{QR3}	**58.28**	**70.85**	**54.84**	**44.30**
GRU	L_q^{QR1}	48.72	61.81	47.23	38.38
	L_q^{QR2}	**60.10**	73.15	**61.81**	44.30
	L_q^{QR3}	59.89	**73.39**	59.14	**45.48**
CNN	L_q^{QR1}	48.02	59.16	46.26	37.02
	L_q^{QR2}	**56.75**	**70.85**	55.91	**43.33**
	L_q^{QR3}	56.33	69.87	**56.24**	42.18

Table 4. Performance on different types of models with different loss functions.

Table 5 shows statistically significant results of the models for each type of network. The maximum pooling is significantly better than other pooling techniques for all of the models. For the RNN-based models we see that the last pooling technique must be avoided since it decreases performance dramatically.

The bidirectional RNN models show their robustness by outperforming the single directional ones. Great improvements in the performance are especially obtained moving from the GRU-based model to the bidirectional-GRU-based one, which has comparable performance with the more complicated RCNN model reported in Lei et. al. 2016 [44].

	# θ	Pool	Test MAP	Test MRR	Test P@1	Test P@5
BM25			55.97	68.03	53.76	42.46
CNN	400K	mean	51.33	64.43	49.04	39.33
	400K	max	**56.75**†	**70.85**†	**55.91**†	**43.33**†
LSTM	423K	last	53.01	65.01	51.08	40.01
	423K	mean	55.62†	69.11†	54.38†	42.57†
	423K	max	**57.73**†	**70.24**†	53.75†	**44.19**†
Bi-LSTM	846K	max	**58.44**	**72.14**	**58.33**	43.01
GRU	404K	last	56.17	68.17	54.84	41.47
	404K	mean	55.92†	69.72†	55.38†	41.94†
	404K	max	**60.10**†	**73.15**†	**61.81**†	**44.30**†
Bi-GRU	481K	max	**61.51**	**74.22**	61.29	**45.65**

Table 5. Reported average results over 5 runs, with a batch size of 40 queries. The second column denotes the number of trainable parameters and the third column indicates the pooling technique. These results are obtained with pretrained word embeddings on a language model left constant throughout training and dropout probability 0.2. Hidden dimensions are 667, 240, 280, 400 and 400 for CNN, LSTM, GRU, Bi-LSTM and Bi-GRU, respectively (kept constant as found in [44]).The bidirectional models use 200-dim for the forward pass and 200-dim for the backward pass and their resulting representations are concatenated.

It should be noted that the models where the word embeddings were pretrained by the language model and kept as constants perform better than: a) the ones utilizing no pretraining in the word embeddings and b) slightly better than the models with pretrained word embeddings left as parameters for fine tune. This is consistent for all type of networks, with the CNN-based having the least effect, and the GRU-based having the greatest effect, with statistical significance only in the last case.

It is worth mentioning that the CNN-based model has a basic performance and is slightly worse than the performance reported in [44]. This might be due to the use of narrow convolution instead of a wide one, or in the use of different activation functions. On the contrary, in research publications running the Semeval 2016-17 task 3 competition, it was reported that CNN models outperform the RNN ones, which is surprising in NLP. Since this is not the case in our domain, a possible reason could be that questions in QatarLiving (Semeval domain) are mainly short questions, therefore the convolutions are able to capture the whole question's structure and semantics; which is not the case in the Askubuntu questions that can be really long.

Neural Network based ranker compared to BM25 model

Overall, the neural network-based approach explained in section 4.1 beats the conventional BM25 model; however, there are cases where the later performs better than the former. We take a closer look on the failures of BM25 where the NN-based model beats this method significantly, or performs even worse.

An investigation of the results has shown that the neural ranking model performs worse than the BM25 in some cases mainly due to one of these reasons:

- All of the candidate questions have similar vocabulary, but the similar questions either contain some words describing the exact specific topic, and these words appear rarely in the corpus; therefore the BM25 model gives high scores to the correct questions because that word has high idf value (e.g. "ifolder").
- All the candidate questions capture the main topic of the query, but due to over-generalization by the network, the ranker does not recognize which questions are better matches, i.e. which candidates contain the exact same topic. Whereas the BM25 model recognizes that the similar questions contain more identical words to the dissimilar ones, and even if the words are of low importance for the NN-based raker they get higher scores in BM25.

  For example, the NN-based model can learn to match the lexically different "high ram eaters" and "already 200mb on swap", while ignoring the topic difference of the two questions talking about a laptop and a server, respectively.

These problems can overlap, but the second one is the main problem of deep neural network based solutions, i.e. the models learn to generalize and they fail when we want them to be more specific.

6.1.1 Evidence supporting the tags utilization

To validate whether performance of the Question Retrieval task is improved, and whether tags utilization might be useful for further improvements, we investigate the test queries and divide them into the following categories:

- ***Queries with potential room for direct or indirect improvement with the tags utilization.***

  The "direct improvement" cases includes queries that have common tags with (almost) only their similar candidates –see the example in table 6.

  The *"indirect improvement"* case includes queries where either all the candidates have some of the query's tags but (almost) only their similar candidates have a lot of common tags, or dissimilar candidates have no common tags with the query's tags. Ideally, when one has this knowledge, he might score these questions lower, giving space for the similar questions to be ranked higher.

  The *"indirect improvement"* category might contain examples where some of the query's and the similar questions' tags do not appear in the title or body of the questions. If the model learns the correlation of the text with the tags, the probability to identify topics referred in the text when not implicitly mentioned is higher. Moreover, if the model knows all the topics addressed in the corpus, it might be easier to decide the similarity of the questions when their tags are known. An example for this subcategory is depicted in table 7.

- ***Queries where no potential room for improvement is assumed with the tags utilization.***

  This category is composed by queries where only some of the most frequent tags exists in the query; in other words, the tags appear in almost every candidate and are not indicative of the exact problem that separates the similar questions from the dissimilar ones (example in table 8). Other examples in the category can be queries that have no common tags with their similar questions, and overall the candidates seem to have a variety of tags.

Query	how to disable networking from command line without sudo ? i need to ssh to localhost using root account , by ssh root @ localhost . when it prompts for passwords , i can not login with all possible passwords . on setting of localhost machine , regular user xxx and root user share the same password (the password that works for sudo -s) , but it does not works for ssh root @ localhost . so where to look at the password for ssh root @ localhost ps : with the password , i can login to regular account on ssh xxx @ localhost . pps : to answer further questions on motivation of doing so , localhost is just a computer in a private network and setting up ssh root @ localhost is just to relieve some manual management in a prototype system . **ssh**, **root**
Retrieved	why phpmyadmin did n't work ? when run phpmyadmin it is not working and i get this message : not found the requested url /phpmyadmin/ was not found on this server . apache/2.2.22 (ubuntu) server at localhost port 80 how it 's not found in case of i 'm just setting up from few minutes . and i just use the commands in this link http : //www.howtoforge.com/ubuntu_lamp_for_newbies without this comand mysql -u **root** and mysql > set password for 'root ' @ 'localhost ' = password ('yourpassword ') ; because i get this error error 1045 : access denied for user : 'root @ localhost ' (using password : no) and i do n't know about that command because it 's first deal with ubuntu installation, mysql, lamp, phpmyadmin
Similar	permission denied for **root** @ localhost for ssh connection i just installed ubuntu 14.04 and lamp on that . then , i wanted to configure my server , so tried out this tutorial . when i give the command : ssh root @ localhost i get : permission denied , please try again . i have logged in as **root** user through the command : sudo -i i also tried the same , by logging in through : sudo -s i use the same password as that i used to log in as user , but still am getting the same error message . could someone help me out here ? ps : i looked into this question but did n't seem to work for me . server, permissions, **ssh**, apache2

Table 6. Question from the AskUbuntu forum, as ranked by a the BM25 method. This is an example of the "room for direct improvement" category. The full input and output of the model is given in Appendix A

Query	possible **memory leak** on ubuntu 10.04 lts i am having an memory issue with one of my ubuntu servers which is running as a vm through vmware esxi . the servers primary function is an email server , running sendmail and dovecot , with squirrelmail and roundcube as the web clients . it also runs our staff intranet page using wordpress . we have approximately 50 users on this page at any given time . the same goes for the email clients . i have been having issues with it locking up since i upgraded to 10.04 from ubuntu 8 hardy . if i reboot the server , (i have 3gb allocated to this vm) , the memory is freed up , but starts to slowly get more and more used up . this will happen until the server locks up , which is obviously ideal . here is a screen capture of my htop command : here is my free -m command : the two show different percentages , so i ’m confused on what to do next . performance
Retrieved	what does cached memory mean when viewing htop ? while viewing the reports of htop , i would like to know what the orange/brown ’cached memory ’ bars actually indicate . really i ’m looking for a more practical explanation of what i ’m looking at , rather that pure cs terms . though i ’d like to have both . when i see a large amount of memory being (having already been ?) cached , will that have any effect on system performance ? what would normally cause the cached memory to increase so much while the used memory (green) stays pretty low ? here is a screen capture of an example htop report ram, cache, htop
Similar	how to detect a memory leak ? i seem to have a larger memory leak on my current ubuntu system after reporting strange eclipse memory errors (eclipse : constant , different out of memory errors) i started to get ’not enough memory ’ error messages in my console today - while doing simple tasks like typing in sudo -s - or even - free -m typing in ’free -m ’ repeadetly showed me how my ram quickly goes up from 700m to 900m , growing up to the size of 2000m in a few seconds (after freeing up memory with echo 3 > /proc/sys/vm/drop_caches) eclipse isnt the cause , i completly killed the process and the ram still was going up . is there any way to detect where the leak is coming from ? i cant even update my system anymore , since apt-get update fails (probably because it ’s out of memory) using ubuntu 11.10 **memory-leak**

Table 7. Question from the AskUbuntu forum, as ranked by a the BM25 method. This is an example of the “room for indirect improvement” category. The full input and output of the model is given in Appendix A

Query	how to upgrade 12.4 to 13.10 i ’m tempted to upgrade my desktop running 12.4 to 13.4 as 13.4 runs really well on my laptop . can i do this without upgrading to 12.10 first (want to avoid fresh install) or is there no way arround it ? thanks in advance , jowan 12.04, **upgrade**, 13.04, lts
Retrieved	upgrading beta to full version work without bugs ? when the beta version of 13.4 comes out , i would like to install it and therefore put all my programs , files , and data on it . on the 18th when the original version of 13.4 comes out , will i be able to upgrade the beta into the original without any issues and successfully run it without bugs . i ’m asking this because when i upgraded 12.4 to 12.10 it had a lot of glitches to it . will the 13.4 run the same after upgrading as if i was to install the it directly as it is ? **upgrade**, beta
Similar	how to upgrade ubuntu from 12.10 to 13.4 without lossing anything ? well title says all most everything . i am using ubuntu 12.10 . installed using windows installer . is there anyway to upgrade it ? any how much data i will lose ? i have installed some softwares and there is some docs . **upgrade**

Table 8. Question from the AskUbuntu forum, as ranked by a the BM25 method. This is an example of the “no room for improvement” category. The full input and output of the model is given in Appendix A

Continuing our comparison of the the BM25 and the NN-based model, we see the performance of the two models on the above categories in figure 13. The figure indicates that not only the queries hypothesized to be improved by the tags-usage have been improved, but unexpected queries as well. This indicates that the neural network based ranker is actually able to capture many latent topics (probably corresponding to the tags) from the questions' descriptions, and this was an expected result. Our main remaining goal for this work is to further improve the performance of the model on –at least– the first category.

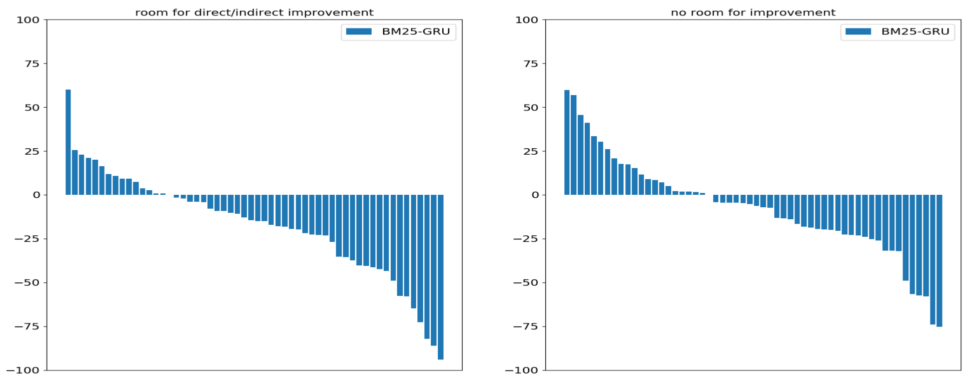


Fig. 13. Mean Average Precision distribution of BM25-GRU per query. (A positive value means BM25 achieved better than the GRU model for a query.)

6.2 Neural Network Based Tags Recommendation Model

In table 9, we see the existing approaches for tags recommendation mentioned in section 3.3 which have been applied on data of the same forum (but on a different dataset). We note that in opposition to our method, none of these approaches is a mere Neural Network based method, neither addresses the whole corpus of tags without pre-processing.

	# objects	# tags	P@5	P@10	R@5	R@10
EnTagRec	37354	243	0.3580	0.1930	0.8150	0.8760
TagCombine	37354	346	-	-	0.5686	0.7273
EnTagRec+	37354	346	0.3610	0.1960	0.7370	0.8700
This Work	167765	908	0.3398	0.20.05	0.74.11	0.8458

Table 9. Results of previous research on AskUbuntu forum's data.

A fair comparison can not be done with the performances reported in previous work since the dataset differs. Therefore, we test the TagCombine method and Logistic Regression as binary relevance on our data. Table 10 gives a comparison of the models using cross validation. Surprisingly, the Logistic Regression performance is very close to the NN-based approaches 10. Here, the hidden dimensions used in LSTM and GRU models were 100, and 300 for the CNN-based model, with a MLP of 100 hidden dimensions, since we noticed that more complex models (e.g. with the dimensions used in QR task) fall in local optima that fail to exceed Logistic regression's performance.

	P@1	P@5	P@10	R@1	R@5	R@10	F@1	F@5	F@10
TagCombine	37.73	24.30	15.80	18.08	54.23	68.89	24,45	34.19	25.71
Logistic Regression	67.11	32.79	19.14	33.19	71.76	81.96	44,41	45,01	31.03
CNN	**70.92**	**33.98**	**20.05**	**35.09**	**74.11**	**84.58**	**46.94**	**46.59**	**32.42**
LSTM	70.79	33.86	19.78	34.88	73.96	84.10	46.73	46.45	32.03
GRU	69.13	33.64	19.50	34.15	73.74	83.75	45.72	46.20	31.63

Table 10. Results with 5-fold cross validation. We use a batch size of 128 questions. These results are obtained with pretrained and static word embeddings on a language model and 0.2 dropout probability.

NN-based model versus TagCombine

Figure 14 shows the difference on Mean Average Precision per question between the TagCombine and the GRU-based model for the questions where TagCombine fails [7] Cases below -75 in the graph are questions with few tags –usually one– where TagCombine recommends the relevant tags somewhere between rank 5 and 10, whilst the neural model ranks them on top of the list. Cases lying close to -5 in the graph are mostly questions with $3-5$ tags where the model fails to rank all of them on top. The neural network model seems to learn how to focus on key words in the text, and whether some successive words define a topic, in contrast to TagCombine.

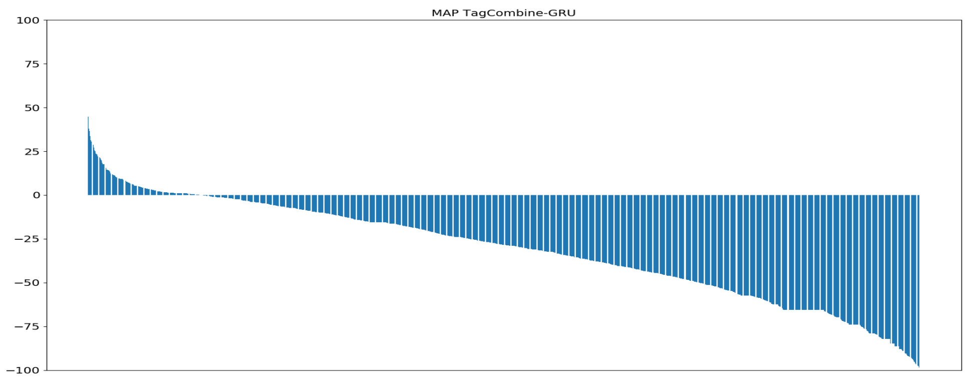


Fig. 14. $MAP_{TagCombine} - MAP_{GRU}$ distribution for the failing cases.

NN-based model performance

In table 11, the evaluation based on the same split with the Question Retrieval task is made. While previous work does not consider the Mean Average Precision here it is reported as the most important metric since every question contains at most 5 tags, and precision at rank 5 is impossible to be 100%. Considering the amount of tags each question has, the upper bound of P@5 is calculated and shown in table 11. The NN-based models perform very well. Equal performance is obtained with dropout probabilities in [0.1, 0.2, 0.3] and either with a 1 hidden layer MLP at the end of the network or a single layer of perceptrons the performance does not change. The results of table 11 give a positive sign to use the NN-based models as a source of extra knowledge in the question retrieval task.

Again the pretrained word embeddings on a language model used as constants of the model give $\approx 1.5\%$ higher MAP value in all network types, than a random initialization or a fine tuning of the pretrained embeddings.

	Test P@5 (47.16)	Test R@5	Test F1@5	Test MAP
CNN	34.16	75.31	46.97	66.87
LSTM	**34.29**	**75.62**	**47.18**	**67.09**
GRU	33.96	75.13	46.78	66.69

Table 11. Results on constant sets –same with the QR datasets, averaged over 3 runs. We use a batch size of 128 questions. Hidden dimensions are 240, 280 and 667 for LSTM, GRU and CNN, respectively (as in QR task).

6.3 Question Retrieval Model pre-trained on the Tags Prediction task

Table 12 shows results obtained when the QR model is first trained on the tag recommendation task as explained in section 4.3. The suffix '_LM_TP' denotes the model with constant word embeddings trained by a

[7] A failure is defined if $MAP \leq average\_MAP$ or $R@10 \leq average\_P@10$ and a success is defined if $MAP \geq 75$.

Language Model, and initialized Encoding Layer's parameters trained on the tag recommendation task. Similarly, '_TP_TP' denotes the model with constant word embeddings trained on the tags recommendation task as well as initialized Encoding Layer's parameters trained on the same model.

Overall, pretraining the QR model on the tags recommendation task does not make any statistically significant improvements, neither damages. However, we can see that the CNN-based model performance gets fairly increments.

Literature for pretraining method, mostly concerns supervised tasks with insufficient data, and aims to pretrain the model on an unsupervised task. This has led to the conclusions that such pretraining can work as a good initialization of the parameters, since they lie in a good spot on the error surface of the unsupervised task and compress the data in a way that helps other tasks to learn. This has been shown in the work of Lei et. al. 2016 [44], by pretraining the NN-based question retrieval model on an auto-encoder model.

In our experiments, the fine tuning of the main task takes at most 4 epochs, in opposition to the main task without pretraining that takes at least 15 epochs to optimize. This might indicate that initialized parameters lie close to a local optimum of the question retrieval task error. However, since results do not support performance increments, this local optimum is not a sufficient minimum point. Table 12 does not support that pretraining the model on tags recommendation can lead to better question retrieval models.

Our hypothesis is that the task of tags recommendation that forms a supervised task relies on different data statistics. The task consists of imbalanced targets and as we have seen in previous sections, different features and techniques are preferred when modeling the individual tasks. Another possible explanation can be that our networks are not complex enough to need the weights initialized by the TR model. In Erhan et. al. 2010 [23] we see that the pretraining technique becomes more needed when the network layers increase.

	Test MAP	Test MRR	Test P@1	Test P@5
CNN	56.75	70.86	55.92	43.33
CNN_LM_TP	**58.07**	**72.99**	**58.60**	43.44
CNN_TP_TP	56.59	69.92	54.84	**44.41**
LSTM	57.73	70.24	53.77	44.20
LSTM_LM_TP	**58.29**	**71.17**	**57.17**	44.27
LSTM_TP_TP	56.46	69.14	55.38	**44.62**
GRU	**60.10**	**73.15**	**61.82**	44.31
GRU_LM_TP	58.88	72.48	58.60	**44.91**
GRU_TP_TP	57.71	70.34	57.30	43.44

Table 12. Results of the QR model pretrained on the TP task. The first three rows are copied from table 5 which shows the simple models' performance.

6.4 Question Retrieval and Tags Prediction jointly trained

The results of tag recommendation in section 6.2, shown that a neural network is able to model the task equally successful either with a 1-Layer MLP or an SLP after the input and encoding layers. Here, we start our experiments with the join model (explained in section 4.3.2) consisting of the encoder shared across both tasks and a single layer perceptron used only for the TR task.

This architecture gave results with $\approx 2\%$ decreased MAP. However, the result is not surprising since literature supports that the probability of *negative transfer* increases when all parameters are being shared.

We then replace the Single Layer Perceptron by a 1-hidden-MLP and we see the models' performance dramatically increased. This is in agreement with the work in Hashimoto et. al. 2016 [31] and Collobert et. al. 2011 [20], who support that multiple tasks performed better when performed in different layers. The results are shown in table 13.

	Test MAP	Test MRR	Test P@1	Test P@5
CNN	56.75	70.85	55.91	43.33
CNN_TP	**58.32** †	**74.49** †	**61.10** †	**44.95** †
LSTM	57.73	70.24	53.76	44.19
LSTM_TP	**58.91**	**73.73**	**60.28**	**45.27**
GRU	60.10	73.15	61.81	44.30
GRU_TP	**60.59**	**74.87**	**61.82**	**45.37**

Table 13. Results on the Question Retrieval task of the baseline neural network models and joint learning models (denoted by '_TP'). Equivalent results are obtained with $\alpha \in \{0.6, 0.7\}$ and $\beta \in \{0.3, 0.4\}$, either with a pre-trained model on the QR task or not.

Fig. 15 shows results of the joint learning framework for the cases we introduced in section 6.1.1. In a successful application of the joint learning, the yellow color must be mostly below zero in the graph named "room for direct/indirect improvement", while the graph named "no room for improvement" can have arbitrary yellow and blue distributions. In the joint learning framework, towards a better Question Retrieval model, the CNN-based model is the most successful one, according to table 13 and fig. 15, followed by the LSTM-based model, which shows some improvement and the GRU-based model where improvement is not clear. In other words, the knowledge transfered from the tags recommendation model under this framework, is higher for the least strong models, which could not capture good representations when trained only for their task. Notice that the performance of the three NN-based models on the question retrieval task becomes similar to the performance of the Bi-GRU ranking model. We suspect that the bidirectional RNN-based models will get little or no improvements from the joint learning framework.

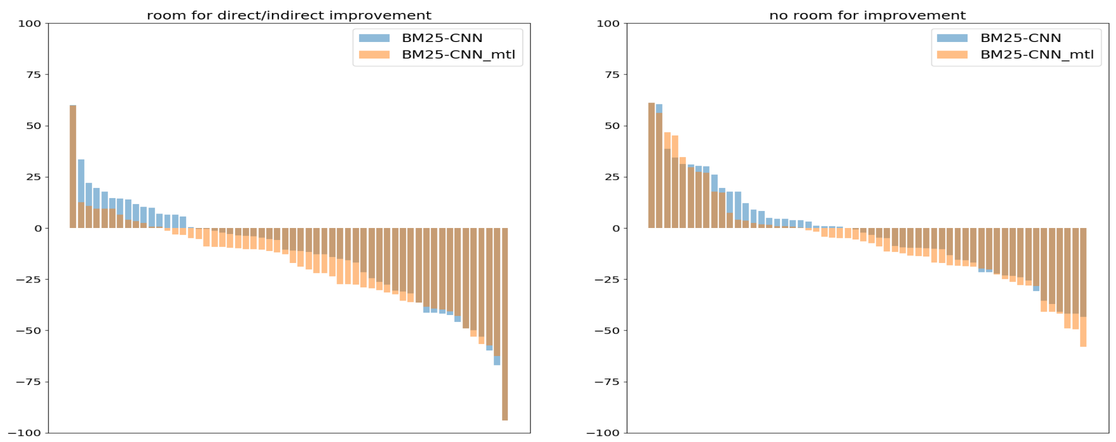
We take a closer look to the CNN-based single retrieval model and the CNN-based multi task model. The cases where the proposed model performs significantly better are:

- queries with one or few similar questions with common tags –with the query tags– and dissimilar questions with or without common tags. The new model ranks the similar question on top –in the case of one similar–, or all the similar questions higher than the previous model.
- queries with a similar question with no common tags, but with text that refers on query's tags. The new ranker scores this question on top.

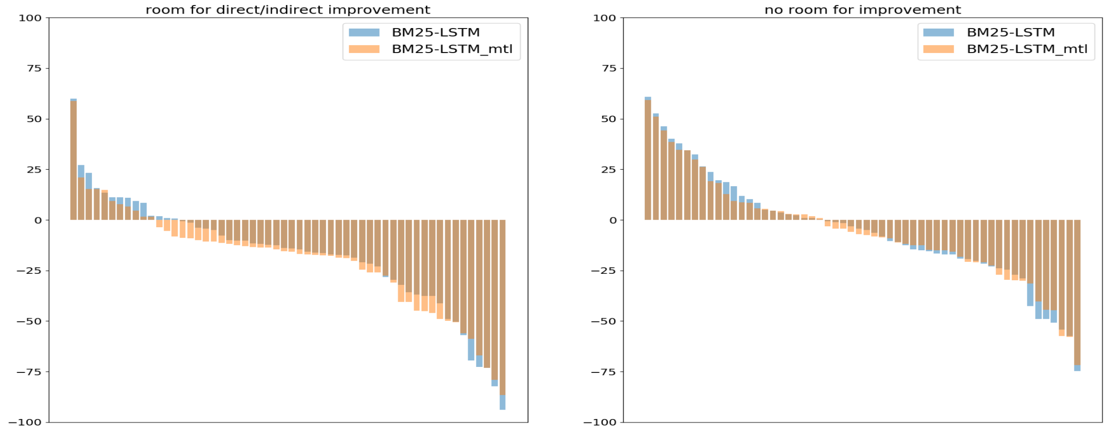
Similarly, the cases where the proposed model performs slightly better than the single QR model are:

- queries with one or many similar questions, referring to unfamiliar topics, i.e. questions with rare tags. The new model ranks most similar questions slightly higher than the previous one.
- queries with one or many similar questions, referring to very famous problems, i.e. questions whose tags appear often in the corpus. The new model ranks most similar questions slightly higher than the previous one.

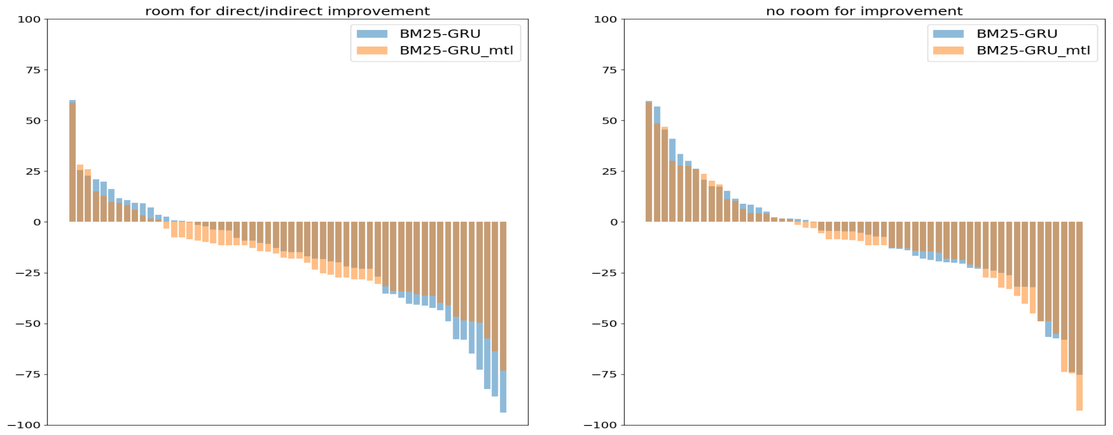
These indicate that the model pays attention on the tags, but not in an explicit way; instead, it encodes the text in a way that summary of the important topics mentioned in the forum can be obtained. Therefore, some of the difficult cases for the model are the very common and very rare queries.



(a)



(b)



(c)

Fig. 15. Compared performance on the Question Retrieval task for the joint learning and the simple question retrieval models. The lines indicate performance_of_BM25 - model_performance on a single query. The cases are split in the two categories introduced in section 6.1.1.

In table 14 results of the TP task [8] are shown at the point where the models were stopped from the training procedure. The training is stopped after 5 hours of training due to limited sources. However, in cases where the learning was not interrupted, the Tags Recommendation task continued improving resulting to a decrease of ≈

[8]These results are based on tags with $20 \leq tag\_occurrences \leq 5000$ and removal of the tags '12.04', '14.04', '11.10' etc. that indicate any version and found empirically uninformative.

2% MAP, while the Question Retrieval performance has stopped improving and started decreasing. Considering that multi task learning approaches usually apply early stopping on each task individually [13], and that our experiments where using $\alpha \geq \beta$, we suspect that the Tags Recommendation task can also be improved by this framework. Few experiments on TR models pretrained by the QR task show support to our belief, however due to limited time sources we could not give any conclusions and significant results on this.

	Test P@5	Test P@10	Test R@5	Test R@10	Test MAP
CNN	**26.21**	**14.65**	**76.66**	**84.30**	**67.45**
CNN_TP	25.23	14.19	75.17	83.15	65.56
LSTM	**25.89**	**14.4**	**75.36**	**83.9**	**66.9**
LSTM_TP	25.39	13.33	74.86	83.23	65.03
GRU	**26.43**	**14.84**	**77.10**	**85.40**	**67.37**
GRU_TP	25.36	14.24	75.04	82.52	66.18

Table 14. Results of the baseline and the joint learning models on the Tags Prediction task. Equivalent results are obtained with $\alpha \in \{0.6, 0.7\}$ and $\beta \in \{0.3, 0.4\}$, either with a pre-trained model on the QR task or not.

7 Conclusions and Future Work

7.1 Main Conclusions

This work provides a method for Question Retrieval in Community Question Answering utilizing deep learning models and an auxiliary task: Tag Recommendation. The idea of using the topic categories in order to model text similarities is not new, however, the joint learning of Tag Recommendation and Question Retrieval within a neural network framework for modeling QR has not been explored before –at least on the AskUbuntu forum data that we use.

Firstly, we have conducted several experiments to model Question Retrieval using neural networks, using data from the AskUbuntu forum. We found the GRU and Bidirectional GRU based models outperforming the CNN, LSTM and Bi-LSTM based models, while experiments on three variations of the max-margin loss function supported the statement that information retrieval tasks are best modeled with loss functions defined on the query level and not on the pairs of questions.

Moreover, we investigated the results of the BM25 model and NN-based models, and saw that the neural ranking models can generalize better, with the price of failing when most of the candidate questions ask about the same generic problem but specific keywords indicate their differences. To approach the problem of generalization, we hypothesized that tags assigned to the questions can be a source of topic disambiguation, and can give confidence to the question retrieval model about which questions are similar on the exact topic and which are only sharing a general problem.

In order to give the information of tags to the neural ranking model, we suggested to either pretrain the model on tag recommendation or jointly train it on our main task, question retrieval and the auxiliary task, tag recommendation. In either case, the task of tag recommendation should be firstly tested on a model of similar architecture to ensure such model can learn useful representations that recognize the problems addressed in the questions.

For that, we conducted several experiments and obtained neural network models that outperform the state of the art systems proposed for tags recommendation in Stack Overflow forums. Additional outcome from our work was that our main and our auxiliary task prefer different features and encoding techniques since the former performs the worse with a CNN-based model, whereas the latter performs the best.

Our last experiments were on the combination of the two models. Pretraining the QR model on tags recommendation showed no clear trends; The CNN and LSTM based models performances were boosted, but the GRU based model had negative effects. This implies that the initialization of QR weights according to 'what features make representations useful for TR' can lie close to a minimum point of the question retrieval error surface that is not optimal, and can act as a negative transfer of knowledge.

In opposition to the pretraining technique, the joint learning of the two tasks shown positive outcomes. The experiments showed that if we use a model consistent of the encoder shared among both tasks and a stacked MLP used for tags recommendation, our main task get performance increments. Our experiments supported the statement that different tasks should be predicted on different layers in order to avoid negative transfer. Additionally, the weaker models, obtained higher improvements by the sharing of knowledge than stronger ones.

Although tags assigned to the questions define very generic topics that are addressed in the forum, or very specific ones, the overall use of a NN-based model to model both QR and TR is successful. The model learns to make better representations for the questions, that are correlated to the main topics addressed in the forum, and therefore, it can make better decisions about their similarity. We also conclude that a context-based model that learns to identify similar questions is able to predict the questions' tags at the same time.

7.2 Recommendations for future work

In a similar work of Lei et. al. 2016 [44], the pretraining of the neural ranker gives none or little performance to the CNN and RNN based models, while it dramatically increases the performance of the non-consecutive convolutions (RCNN) based ranker. Therefore, it would be nice to try our experiments on the RCNN model as well.

Furthermore, it would be interesting to compare our joint learning method with a neural ranking model that uses region embedding [39]. Region embeddings map text regions to high-level concepts, in contrast to the word embeddings that learn individual words in isolation. Therefore, it would be nice to see if region embedding is enough to identify tags as latent topics and give similar performances without the use of tags.

Moreover, we would like to use correction rules and re-annotate the questions' tags to fix the missing and incorrect labels, and test our proposed method on more forums. We are also interested at more experimentation on various architectures concerning the joint model, and finding the one that increases the baseline performance on both tasks at the same time.

A different way to address the problem of tag disambiguation occurring by general and specialized tags could be to represent tags as multi-dimensional vectors and model both questions and tags in the same or collaborative hidden spaces. Such tags' encodings, could get extra knowledge from Wikipedia descriptions about them, and their combination with question retrieval might introduce a clearer communication between the two tasks.

Last but not least, the attention mechanism [75] used in an increasing amount of literature, could be exploited for a clearer investigation of what our models focus on and could give a clearer analysis of what the proposed method can achieve.

References

[1] Shivani Agarwal and Michael Collins. *Maximum Margin Ranking Algorithms for Information Retrieval*, pages 332–343. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[2] Everton Alvares Cherman, Jean Metz, and Maria Carolina Monard. *A Simple Approach to Incorporate Label Dependency in Multi-label Classification*, pages 33–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[3] Peter Babinec and Ivan Srba. Education-specific tag recommendation in CQA systems. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization - UMAP17*. ACM Press, 2017.

[4] Fabiano M. Belém, Jussara M. Almeida, and Marcos A. Gonçalves. A survey on tag recommendation methods. *J. Assoc. Inf. Sci. Technol.*, 68(4):830–844, apr 2017.

[5] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, mar 1994.

[6] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[8] Dasha Bogdanova, Cícero Nogueira dos Santos, Luciano Barbosa, and Bianca Zadrozny. Detecting semantically equivalent questions in online user forums. In *CoNLL*, 2015.

[9] Daniele Bonadiman, Antonio Uva, and Alessandro Moschitti. Multitask learning with deep neural networks for community question answering. 02 2017.

[10] Klaus Brinker, Johannes Fürnkranz, and Eyke Hüllermeier. A unified model for multilabel classification and ranking. In *Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy*, pages 489–493, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.

[11] Xin Cao, Gao Cong, Bin Cui, and Christian S. Jensen. A generalized framework of exploring category information for question retrieval in community question answer archives. In *Proceedings of the 19th International Conference on World Wide Web*, pages 201–210, New York, NY, USA, 2010. ACM.

[12] Benjamin A. Carterette. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Trans. Inf. Syst.*, 30(1):4:1–4:34, March 2012.

[13] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, Jul 1997.

[14] Rich Caruana, Shumeet Baluja, and Tom Mitchell. Using the future to ”sort out” the present: Rankprop and multitask learning for medical risk evaluation. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 959–965. MIT Press, 1996.

[15] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, G.s Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems. pages 7–10, 09 2016.

[16] Everton Cherman, Jean Metz, and Maria-Carolina Monard. Incorporating label dependency into the binary relevance framework for multi-label classification. 39:1647–1655, 02 2012.

[17] Kyunghyun Cho, B van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. *On the properties of neural machine translation: Encoder-decoder approaches*. 2014.

[18] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. *Empirical evaluation of gated recurrent neural networks on sequence modeling*. 2014.

[19] Amanda Clare and Ross D. King. *Knowledge Discovery in Multi-label Phenotype Data*, pages 42–53. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[20] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, nov 2011.

[21] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. Neural ranking models with weak supervision.

[22] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 681–687, Cambridge, MA, USA, 2001. MIT Press.

[23] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.*, 11:625–660, March 2010.

[24] Umer Farooq, Thomas G. Kannampallil, Yang Song, Craig H. Ganoe, John M. Carroll, and Lee Giles. Evaluating tagging behavior in social bookmarking systems: Metrics and design heuristics. In *Proceedings of the 2007 International ACM Conference on Supporting Group Work*, GROUP '07, pages 351–360, New York, NY, USA, 2007. ACM.

[25] Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. Applying deep learning to answer selection: A study and an open task. In *ASRU*, pages 813–820. IEEE, 2015.

[26] Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. In *SemEval@NAACL-HLT*, 2016.

[27] Daniel Gardner and David Nichols. Multi-label classification of satellite images with deep learning.

[28] Youyang Gu, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Learning to refine text based recommendations. 2016.

[29] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. Social media recommendation based on people and tags. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 10, pages 194–201, New York, NY, USA, 2010. ACM.

[30] Zellig S. Harris. Distributional structure. *¡i¿WORD¡/i¿*, 10(2-3):146–162, 1954.

[31] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. 11 2016.

[32] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 820–828, 2016.

[33] James Henderson. Discriminative training of a neural network statistical parser. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[34] Francisco Herrera, Francisco Charte, Antonio J. Rivera, and María J. del Jesus. *Multilabel Classification*, pages 17–31. Springer International Publishing, Cham, 2016.

[35] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, nov 1997.

[36] James Hong and Michael Fang. Keyword extraction and semantic tag prediction. 2013.

[37] Yongshuai Hou, Cong Tan, Xiaolong Wang, Yaoyun Zhang, Jun Xu, and Qingcai Chen. Hitsz-icrc: Exploiting classification approach for answer selection in community question answering. In *SemEval@NAACL-HLT*, 2015.

[38] Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. Extractive summarization using multi-task learning with document classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100. Association for Computational Linguistics, 2017.

[39] Rie Johnson and Tong Zhang. Semi-supervised convolutional neural networks for text categorization via region embedding. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 919–927. Curran Associates, Inc., 2015.

[40] Gal Yarin Kendall, Alex and and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. 05 2017.

[41] Gakuto Kurata, Bing Xiang, and Bowen Zhou. Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence, 01 2016.

[42] Gakuto Kurata, Bing Xiang, and Bowen Zhou. Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence. pages 521–526, 01 2016.

[43] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on Machine Learning, PMLR*, 2(32):1188–119, 2014.

[44] Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi S. Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez i Villodre. Semi-supervised question retrieval with gated convolutions. In *HLT-NAACL*, 2016.

[45] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 282–289, New York, NY, USA, 2003. ACM.

[46] Marek Lipczak, Yeming Hu, Yael Kollet, and Evangelos Milios. Tag sources for recommendation in collaborative tagging systems. In *Proceedings of the 2009th International Conference on ECML PKDD Discovery Challenge - Volume 497*, ECMLPKDDDC'09, pages 157–172, Aachen, Germany, Germany, 2009. CEUR-WS.org.

[47] Zachary C. Lipton. A critical review of recurrent neural networks for sequence learning. 06 2015.

[48] Huawen Liu, Shichao Zhang, and Xindong Wu. Mlslr: Multilabel learning via sparse logistic regression. *Information Sciences*, 281(Supplement C):310 – 320, 2014. Multimedia Modeling.

[49] Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH*, pages 3771–3775, 2013.

[50] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[51] Preslav Nakov, Doris Hoogeveen, Lluís M ' Arquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. Semeval-2017 task 3: Community question answering. 08 2017.

[52] Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. *Large-Scale Multi-label Text Classification — Revisiting Neural Networks*, pages 437–452. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[53] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, October 2010.

[54] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *ACL*, 2017.

[55] Tao Qin, Xu-Dong Zhang, Ming-Feng Tsai, De-Sheng Wang, Tie-Yan Liu, and Hang Li. Query-level loss functions for information retrieval. *Information Processing & Management*, 44(2):838 – 855, 2008.

[56] Marek Rei. Semi-supervised multitask learning for sequence labeling. In *ACL*, 2017.

[57] Steffen Rendle and Lars Schmidt-Thieme. Factor models for tag recommendation in bibsonomy. In *Proceedings of the 2009th International Conference on ECML PKDD Discovery Challenge - Volume 497*, ECMLPKDDDC'09, pages 235–242, Aachen, Germany, Germany, 2009. CEUR-WS.org.

[58] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at trec-3. 1994.

[59] Sebastian Ruder. An overview of multi-task learning in deep neural networks. 06 2017.

[60] Tetsuya Sakai. Web search evaluation with informational and navigational intents. *Journal of Information Processing*, 21(1):145–155, 2013.

[61] Adrian Sanborn and Jacek Skryzalin. Deep learning for semantic similarity. 2015.

[62] Cicero Dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. Learning hybrid representations to retrieve semantically equivalent questions. 2:694–699, 01 2015.

[63] Andreas Savakis. A computationally efficient approach to indoor/outdoor scene classification. In *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 4 - Volume 4*, ICPR '02, pages 40146–, Washington, DC, USA, 2002. IEEE Computer Society.

[64] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2):135–168, May 2000.

[65] Mike Schuster, Kuldip K. Paliwal, and A. General. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 1997.

[66] Abigail See, Christopher Manning, and Peter Liu. Get to the point: Summarization with pointer-generator networks. In *Association for Computational Linguistics*, 2017.

[67] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 373–382, 2015.

[68] R. J. SIMES. An improved bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3):751–754, 1986.

[69] John R. Smith and Chung-Sheng Li. Image classification and querying using composite region templates. *Comput. Vis. Image Underst.*, 75(1):165–174, jul 1999.

[70] Yang Song, Lu Zhang, and C. Lee Giles. Automatic tag recommendation algorithms for social recommender systems. *ACM Trans. Web*, 5(1):4:1–4:31, feb 2011.

[71] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[72] Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. Translation modeling with bidirectional recurrent neural networks. In *EMNLP*, 2014.

[73] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 2007:1–13, 2007.

[74] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. 4, 05 2014.

[75] Baiyang Wang and Diego Klabjan. An attention-based deep net for learning to rank. 02 2017.

[76] Shaowei Wang, David Lo, Bogdan Vasilescu, and Alexander Serebrenik. Entagrec: An enhanced tag recommendation system for software information sites. Sep 2014.

[77] Shaowei Wang, David Lo, Bogdan Vasilescu, and Alexander Serebrenik. Entagrec++: An enhanced tag recommendation system for software information sites. *Empirical Software Engineering*, Jul 2017.

[78] Xin-Yu Wang, Xin Xia, and David Lo. Tagcombine: Recommending tags to contents in software information sites. *Journal of Computer Science and Technology*, 30(5):1017–1035, Sep 2015.

[79] Xin Xia, David Lo, Xinyu Wang, and Bo Zhou. Tag recommendation in software information sites. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pages 287–296, Piscataway, NJ, USA, 2013. IEEE Press.

[80] Wenpeng Yin and year=2016 volume=4 pages=259-272 Hinrich Schütze and Bing Xiang and Bowen Zhou, journal=TACL. Abcnn: Attention-based convolutional neural network for modeling sentence pairs.

[81] Min-Ling Zhang and Kun Zhang. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 999–1008, New York, NY, USA, 2010. ACM.

[82] Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. Phrase-based translation model for question retrieval in community question answer archives. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT 2011, pages 653–662, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[83] Guangyou Zhou, Yubo Chen, Daojian Zeng, and Jun Zhao. Towards faster and better retrieval models for question search. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM 2013, pages 2139–2148, New York, NY, USA, 2013. ACM.

[84] Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. Learning continuous word embedding with metadata for question retrieval in community question answering, 01 2015.

[85] Pingyi Zhou, Jin Liu, Zijiang Yang, and Guangyou Zhou. Scalable tag recommendation for software information sites. *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 00:272–282, 2017.

A Appendix

Example from the ‘direct improvement’ category. q, d_q^+ and d_q^- denote the query, a question that is similar to the query and a dissimilar question, respectively:

q	what is the password for “ ssh root @ localhost “ ? i need to ssh to localhost using root account , by ssh root @ localhost . when it prompts for passwords , i can not login with all possible passwords . on setting of localhost machine , regular user xxx and root user share the same password (the password that works for sudo -s) , but it does not works for ssh root @ localhost . so where to look at the password for ssh root @ localhost ps : with the password , i can login to regular account on ssh xxx @ localhost . pps : to answer further questions on motivation of doing so , localhost is just a computer in a private network and setting up ssh root @ localhost is just to relieve some manual management in a prototype system . **ssh**, **root**
d_q^-	why phpmyadmin did n’t work ? when run phpmyadmin it is not working and i get this message : not found the requested url /phpmyadmin/ was not found on this server . apache/2.2.22 (ubuntu) server at localhost port 80 how it ’s not found in case of i ’m just setting up from few minutes . and i just use the commands in this link http : //www.howtoforge.com/ubuntu_lamp_for_newbies without this comand mysql -u **root** and mysql > set password for ’root ’ @ ’localhost ’ = password (’yourpassword ’) ; because i get this error error 1045 : access denied for user : ’root @ localhost ’ (using password : no) and i do n’t know about that command because it ’s first deal with ubuntu installation, mysql, lamp, phpmyadmin
d_q^-	mysql access denied for user ’root ’ @ ’localhost ’ in my server , everytime i try to access mysql i get the error : access denied for user ’root ’ @ ’localhost ’ (using password : no) when i try mysqladmin -u **root** -p password i get access denied for user ’root ’ @ ’localhost ’ (using password : yes) how can i access mysql in my ubuntu server 12.04 ? 12.04, server, password, mysql
d_q^+	permission denied for **root** @ localhost for ssh connection i just installed ubuntu 14.04 and lamp on that . then , i wanted to configure my server , so tried out this tutorial . when i give the command : ssh root @ localhost i get : permission denied , please try again . i have logged in as **root** user through the command : sudo -i i also tried the same , by logging in through : sudo -s i use the same password as that i used to log in as user , but still am getting the same error message . could someone help me out here ? ps : i looked into this question but did n’t seem to work for me . server, permissions, **ssh**, apache2
d_q^-	unable to create database due access denied i just logged in to ubuntu 12.04 as a normal user (vivek) and opened the terminal to create a database , i wrote : create database hello ; and the error i had is : error 1044 (42000) : access denied for user ” @ ’localhost ’ to database ’vivek’ if i login as **root** user then even mysql is not opening and having the error message : error 1045 (28000) : access denied for user ’root ’ @ ’localhost ’ (using password : no) what to do now . mysql
d_q^-	ssh into shell , get ’error : ca n’t open display : (null) ’ i do n’t want to open an x11 session or what not , just to open up a shell to a pc . i type in terminal : ssh -p 1337 localhost after entering my password , i get the banner , last login , and then the error : error : ca n’t open display : (null) also , if i type in a command after localhost , it works xorg, **ssh**, remote-access
d_q^-	mediawiki installation on 14.04 i have followed this guide https : //www.mediawiki.org/wiki/manual : running_mediawiki_on_ubuntu successfully down to the line mysql_secure_installation at which point it prompted me for a password , i entered one and it said : error 1045 (28000) : access denied for user ’root ’ @ ’localhost ’ (using password : yes) then i decided to hit enter as it again prompted me with : enter current password for **root** (enter for none) : and this gave this error : error 1045 (28000) : access denied for user ’root ’ @ ’localhost ’ (using password : no) . thank you for any help . 14.04, software-installation, php, mediawiki
d_q^+	need to connect as **root @ my-server-ip** with ssh i have a vps with ubuntu 14.04 . i need to login as root to my server over ssh . i set a password for **root** : sudo passwd **root** then enabled it : sudo passwd -u **root** then rebooted my vps : sudo reboot but i still ca n’t connect with the root user . how can i enable ssh login as root ? **ssh**, **root**, vps
d_q^-	ssh : connect to host localhost port 22 : connection refused although this has been discussed , but still i am not able to solve this . here is a quick review of what i did and how i got into the trouble : installed ubuntu 12.10 , installed ssh , sudo apt-get install ssh . the favorite ssh was working fine and connected by using ssh localhost . i removed ubuntu , increased the partition and again installed ubuntu . again i fired sudo apt-get install ssh , it showed me reading package lists ... done building dependency tree reading state information ... done when i given command , ssh localhost , it gave me error as : ssh : connect to host localhost port 22 : connection refused **ssh**
d_q^-	“ ’access denied for user ’root ’ @ ’localhost ’ ” when trying to do commands the other day i was using mysql on my local database fine from the terminal . however , when i rebooted the next day and tried to use it , i get the following error for just about everything related to mysql . mysqladmin : connect to server at ’localhost ’ failed error : ’access denied for user ’root ’ @ ’localhost ’ (using password : no) ’ i have been going at this for a couple hours now .

	mysql
d_q^-	use real browser instead of w3m on ssh connections i am using ubuntu as desktop and server os . when i am logged in to the server via ssh on terminal and have to view a web-page (localhost) on the server i use w3m (w3m localhost) . unfortunately w3m is not that easy to handle , as the page has some big menus and uses jquery . so i am wondering if it is possible to use a browser on my desktop to connect to the server via ssh with a real browser (firefox or chrome) . basically it would require to connect with a browser on my desktop to the server over ssh with username and password , and open on that server localhost . is this possible by default , or are there any add-ons for firefox/chrome ? i would prefer firefox . **ssh**, browser, remote-access, w3m
d_q^-	access webpage through ssh i need to access the ieee xplore , but i do n't have right to download out of institute . i can login into the server of institute via ssh , so how can i access ieee xplore through institute server via ssh ? i have searched solutions , some one answer : ssh -l 8080 : localhost:80 user @ remoteserver and then he says : now , point your local browser to localhost:8080 . it should be forwarded to localhost:80 in the remote server . but i still do n't know how to configure my laptop , im using chrome . i appreciate your help very much ! **ssh**, localhost, html
d_q^+	how to deny **root** ssh login + require ssh key for user ? after reading this similar question i 'm unclear how to require ssh keys for user , while denying ssh access to **root** (key or password) . i 'll use sudo after logging in as a user if i need to be **root** . i know how to require ssh keys : permitrootlogin without-password i know how to disable user 'root ' from logging in : permitrootlogin no i know how to whitelist a user : allowusers user i do n't know how to put this all together (or if it 's even necessary ? - ssh key may be good enough) . i would normally solve an answer like this through trial and error , but i 'm afraid i 'll lock myself out of this server ... server, **ssh**, login, **root**
d_q^-	can not login to ubuntu 13.10 hosted in virtualbox from os x mavericks when i try to log in into ubuntu ssh does n't accept my password (password is correct) : last login : sun oct 27 12:11:34 from localhost macbook : jirka $sshjirka@127.0.0.1password : password : password : permissiondenied(publickey, keyboard - interactive).macbook : jirka$ i set port forwarding in in virtual box setting . what am i doing wrong ? virtualbox, macosx
d_q^+	not able to ssh to localhost while setting up hadoop in my ubuntu 12.10 i am not able to ssh to localhost . hduser @ tanmoy : ssh -vvv localhost openssh_6.0p1 debian-3ubuntu1 , openssl 1.0.1c 10 may 2012 debug1 : reading configuration data /etc/ssh/ssh_config debug1 : /etc/ssh/ssh_config line 19 : applying options for * debug2 : ssh_connect : needpriv 0 debug1 : connecting to localhost [127.0.0.1] port 22. debug1 : connect to address 127.0.0.1 port 22 : connection refused ssh : connect to host localhost port 22 : connection refused can anyone help me on this ? **ssh**, localhost
d_q^+	server asking for password even after adding ssh keys this was working fine but not sure suddenly stopped working . i have added my id_rsa.pub key of localhost to authorized_keys of .ssh folder on server . when i try to login to server (running ubuntu 12.10) from localhsot with ssh account @ ip it asks for password . i am not sure what i am doing wrong since it looks to be pretty straight forward process and it was working fine earlier . any help ? server, **ssh**
d_q^+	how to setup passwordless ssh access for **root** user i need to configure a machine so software installation can be automated remotely via ssh . following the wiki , i was able to setup ssh keys so my user can access the machine without a password , but i still need to manually enter my password when i use sudo , which obviously an automated process should n't have to do . although my /etc/ssh/sshd_config has permitrootlogin yes , i ca n't seem to be able to login as **root** , presumably because it 's not a " real " account with a separate password . how do i configure ssh keys , so a process can remotely login as **root** on ubuntu ? **ssh**
d_q^-	installed mysql but can not login : access denied for user 'xxxxxx ' @ 'localhost ' (using password : yes) installed mysql - but not able to login with the user i created . it gives me the following error . access denied for user 'xxxxxx ' @ 'localhost ' (using password : yes) please help . mysql
d_q^+	how to scp a file (s) from ubuntu virtual machine to remote web host ? i 'm trying to scp some files from my local ubuntu 10.04 virtual machine (running on a windows vista platform) up to my remote web host . ssh is enabled on their end and i was able to login via the command line so i know i 'm using the right port # . here is what i tried : i login in to the web host via ssh shell . then at the command prompt " remoteuser @ remotehost " i type : scp -p port # -r myuserid @ 127.0.0.1 : /fromfolder/ /public_directory/tofolder/ where port # is an actual port # like 22 i was asked for myuserid @ 127.0.0.1 's password but it would n't take the password i normally use with this localhost account . what is the best way to use scp ? do i login the remote host via ssh and then run scp at the command prompt remoteuser @ remotehost $ or should i just run scp from my localhost ? 10.04, virtual, scp
d_q^+	how to change my hostname ? my hostname is localhost , reported by hostname command , and terminal prompt **root** @ localhost : # . how can i change it to mismis.com with mismis alias ? and what is the proper configuration . i confused after reading some articles on web . my /etc/hosts : 127.0.0.1 localhost 127.0.1.1 srv345.myweb.com srv345 178.162.231.61 janstone.mismis.com janstone # the following lines are desirable for ipv6 capable hosts : :1 localhost ip6-localhost ip6-loopback fe00 : :0 ip6-localnet ff00 : :0 ip6-mcastprefix ff02 : :1 ip6-allnodes ff02 : :2 ip6-allrouters

	hostname
d_q^-	hmysql access denied for user ’root ’ while using lamp i ’m just want to install lamp server + phpmyadmin normally in case of when setting up mysql i get this error : error 1045 (28000) : access denied for user ’root ’ @ ’localhost ’ (using password : no) i do n’t know what this is . i do n’t know how to run php code in ubuntu because it ’s my first deal with linux . host installation, lamp, phpmyadminname

Example from the ‘indirect improvement’ category. q, d_q^+ and d_q^- denote the query, a question that is similar to the query and a dissimilar question, respectively:

q	possible **memory leak** on ubuntu 10.04 lts i am having an memory issue with one of my ubuntu servers which is running as a vm through vmware esxi . the servers primary function is an email server , running sendmail and dovecot , with squirrelmail and roundcube as the web clients . it also runs our staff intranet page using wordpress . we have approximately 50 users on this page at any given time . the same goes for the email clients . i have been having issues with it locking up since i upgraded to 10.04 from ubuntu 8 hardy . if i reboot the server , (i have 3gb allocated to this vm) , the memory is freed up , but starts to slowly get more and more used up . this will happen until the server locks up , which is obviously ideal . here is a screen capture of my htop command : here is my free -m command : the two show different percentages , so i ’m confused on what to do next . performance
d_q^-	what does cached memory mean when viewing htop ? while viewing the reports of htop , i would like to know what the orange/brown ’cached memory ’ bars actually indicate . really i ’m looking for a more practical explanation of what i ’m looking at , rather that pure cs terms . though i ’d like to have both . when i see a large amount of memory being (having already been ?) cached , will that have any effect on system performance ? what would normally cause the cached memory to increase so much while the used memory (green) stays pretty low ? here is a screen capture of an example htop report . ram, cache, htop
d_q^-	how to set up a mail server which i can then use with email clients ? i ’m trying to set up a mail server on my ubuntu 12.04 box from linode . the box currently runs apache , mysql and php and runs a few of my websites , and i want to set up an email account like myname @ mydomain.com . in the interest of learning , i ’d like to go through the painful task of setting up my own mail server . i ’ve seen lots of suggestions on how to do so , most involve using postfix and dovecot , but i ’m really struggling to understand : what is the difference is between both ? how to set them up properly to work together ? i just want a mail server running imap which i can use a mail client like thunderbird with . email, postfix, imap, dovecot
d_q^-	memory usage statistics different in free and htop when i run free -m on my command-line , it shows my used memory as show below . when i run htop the used memory shown is very low . why is that ? is it some other kind of representation ? i am new to linux . so i am totally blank on these stuff . total 3 used free shared buffers cached 5863 4980 882 0 903 3025 htop : mem [————— # # # # # *****1076/586] the used memory in free shows 4980mb and in htop shows 1076mb . what do they actually represent ? ram
d_q^-	ubuntu sync using too many resources i ’m running 10.04 lts and the ubuntu-sync process is really using a lot memory and cpu resources , especially as time goes on during the day . when i start out the day , (and do a top) the cpu % is typically 6.0 and % mem 0.8 , but as the day goes these numbers increase until my notebook is all but unusable , at which point i have to either re-boot or kill the process . just today it was using 1.9g of memory . using system monitor right now , and looking at memory usage the mib number for ubuntu-syncdaemon is only going up . same problem on ubuntu 11.04 . ubuntuone get 25 % of my 4gb of memory to synchronize . i ’m using an hp elitebook 8640p basic configuration . i do n’t have the same problem on my dell xps1330 with the same ubuntu version .? 11.04, 10.04, ubuntu-one, sync
d_q^-	system monitor disagrees with ’free -m ’ when i start up my computer , free -m says i have about 1800mb allocated and 150mb free . in system monitor , it says i have about 300mb used and 1700mb free . which one is correct ? i am having problems getting my java app to work and i keep getting out of memory exceptions even when system monitor says i have 1gb+ free . what gives ? 12.04, ram, system-monitor
d_q^+	how to detect a memory leak ? i seem to have a larger memory leak on my current ubuntu system after reporting strange eclipse memory errors (eclipse : constant , different out of memory errors) i started to get ’not enough memory ’ error messages in my console today - while doing simple tasks like typing in sudo -s - or even - free -m typing in ’free -m ’ repeadetly showed me how my ram quickly goes up from 700m to 900m , growing up to the size of 2000m in a few seconds (after freeing up memory with echo 3 > /proc/sys/vm/drop_caches) eclipse isnt the cause , i completly killed the process and the ram still was going up . is there any way to detect where the leak is coming from ? i cant even update my system anymore , since apt-get update fails (probably because it ’s out of memory) using ubuntu 11.10 **memory-leak**
d_q^-	secure shared memory

	in my ubuntu deployment script i have written a function to secure shared memory . for a web server it 's important to secure the shared memory to tighten up security . below you can find the code : echo " tmpfs /dev/shm tmpfs tdefaults , noexec , nosuid 0 0 " >> /etc/fstab when i reboot the web server it gets stuck . i ca n't see where it goes wrong . do you have an idea , please let me know . ram, webserver, shared
d_q^-	will my server reboot itself if the memory fills up ? i just got a vds , all said and done with my website , forums and email server set up along with my teamspeak server . my memory usage is at 60 % when it boots and grows to 70 % after about 3 hours of running . i 'm just curious on what will happen if the memory fills up and would it notify me in some way if it did ? server, ram
d_q^+	memory utilization increased from 7 % to 72 % in 10 mins . in ubuntu 14.04 i had installed 14.04 recently on my servers . but some of the servers showing wired observations . memory utilization increased from 7 % to 72 % in 10 min . top output is not showing single process utilizing more . but free -m shows 5 gb out of 7 gb is used . node is running on these servers . but even after stopping all node process it does not free up memory . only restart of servers will free up some memory . had seen system logs and kern.log . nut nothing is found . 14.04
d_q^-	stopping xfce4 from starting up on boot on ubuntu server 10.04 i installed xfce4 on my ubuntu server 10.04 based server to be able to use a gui remotely through vnc . i installed it using the command sudo apt-get install xubuntu-desktop . however , after that , the xfce login screen appears on the server after a reboot . how can i prevent that from happening ? how can i revert to having the server simple boot into command line ? boot, gui, xfce, xubuntu
d_q^-	how to increase memory allocated to java ? java.lang.outofmemoryerror : java heap space i have an ubuntu server 12.04 using a reverse proxy so that tomcat runs on top of apache . and i have a confluence instance on that machine , and a ticket system . so nothing special . memory is 8gb that 's more than enough for this use . from time to time , and in the last few days more often my confluence page crashes . the log (confluence logs catalina.2014-05-06) shows this : severe : socket accept failed java.lang.outofmemoryerror : java heap space so how can i avoid this ? how can i give that process more memory . java
d_q^-	server locking up router i have an ubuntu server 14.04 with two kvm vm 's running 12 and i am having issues with them locking up my router . i will loose all connectivity ca n't ssh or webmin into them and any external users loses the connection anyone ideas ? one server run 's mumble the other server runs a minecraft server . quit editing the post and answer it . networking, 14.04, server, kvm, router
d_q^-	mount virtualbox sharedfolder in ubuntu vm on boot i have a ubuntu vm running in virtualbox . i have a shared folder set up as myshare i have a folder created in my home directory (˜/jamesw) as host . using the line sudo mount -t vboxsf myshare host from my home directory in terminal mounts the drive correctly , but if i reboot it is not mounted again . this is a vm i use for running a simple web server and i would like this share mounted each time i boot . is there any way to do this ? especially given that it requires sudo ? mount, virtualbox, sudo
d_q^-	top command not clear i am using top command on my centos server . mem : 8055100k total , 3832228k used , 4222872k free , 57968k buffers swap : 8191992k total , 0k used , 8191992k free , 3574740k cached in case of memory , i see 3832228k used , but when i check per process consumption , its showing 0 % . can some one tell which process is consuming this memory . i have also attached a screenshot for this . memory-usage, top
d_q^-	evolution not showing latest email in my imap inbox i 've got an imap mail server running dovecot . i use several different mail clients – k-9 mail on my nexus one , and roundcube webmail when i 'm not on my main computer . on my main computer , i use evolution . however , it does n't show the latest emails on the server . i can see them with other clients ; just not evolution . is there a way to force evolution to re-load the entire folder ? evolution, imap
d_q^-	how to setup sendmail with nginx i have a ubuntu server with nginx i have installed sendmail on it by running apt-get install sendmail i was also able to sendmail using sendmail < my_email_address > hello . the mail i received on the specified email was from myusername @ ns1234.ip-12-34-23.eu . here myusername is the username of my ubuntu machine i want to create new mail accounts like info @ mydomainname.com contact @ mydomainname.com etc . i have read many guides but they assume that the person is running apache2 so the folders were different from what i have nginx, sendmail
d_q^-	upgraded from 8.04 to 10.04 , can not boot i upgraded an old , handed down web server with a heavily customized wordpress website from 8.04 server lts to 10.04 server lts using a guide i found on ubuntu . this seemed to go fine . there were a few errors , but on the whole , the upgrade succeeded , and afterwards i had a working system , web site was still online , good deal . then i rebooted . after the reboot , i am greeted with this error message , on every boot : here is my disk information , which i got by using a linux live boot cd : i spent 8+ hours combing through forums , google , etc yesterday trying to figure this out . i 'm really at a loss . at this point , i am converting a backup of the old physical web server to a vm for a temporary website.. but this one needs to get fixed . boot, server, upgrade, 10.04
d_q^-	how to send mail from the command line using tor ?

	i wanted to send email from my own mail server using a ip from tor network . so , i installed the sendmail and tor in my ubuntu server . but somehow i am not able to make sendmail to use tor network . the command : tsocks sendmail -t someone @ google.com is sending email from my original ip than tor ip . i tried to google the problem and it seems that sendmail dont support socks proxy . thus , is there a way to send mail from any ubuntu mail server over tor using command line ? if yes which and how ? server, mail, tor, sendmail
d_q^-	roundcube not sending email i tried to set up an email server with dovecot-postfix and using roundcube as the webmail application . the problem is that i can see and receive emails inside roundcube but when i try to send one , the recipăžent does n't receive it , although roundcube prompts " message succesfully sent " . i used mutt to check if i can send mail from the server , and is working ok. what could possibly go wrong here ? email, mail-server
d_q^-	how to detect a memory leak system wide ? i 'm using a ubuntu lucid box at work which just runs for ever (almost never shut it down) . since upgrading to lucid i 've noticed that after 2 or 3 weeks w/o rebooting , memory usage becomes huge : ram completely full (3gb) and 6 gb swap . i suspect this is caused by some app (or driver) leaking memory , but i have no clue which one it might be , because memory usage increases very slowly . so , i need a program to detect if any process in my system is slowly leaking memory . what i think would do it is a program which takes a daily measurement of the memory usage of each process (like top or htop) , and shows me the evolution of that usage over the time . googled for it . found nothing . 10.04, memory, top, **memory-leak**

Example from the 'no improvement' category. q, d_q^+ and d_q^- denote the query, a question that is similar to the query and a dissimilar question, respectively:

q	how to upgrade 12.4 to 13.10 i 'm tempted to upgrade my desktop running 12.4 to 13.4 as 13.4 runs really well on my laptop . can i do this without upgrading to 12.10 first (want to avoid fresh install) or is there no way arround it ? thanks in advance , jowan 12.04, **upgrade**, 13.04, lts
d_q^-	upgrading beta to full version work without bugs ? when the beta version of 13.4 comes out , i would like to install it and therefore put all my programs , files , and data on it . on the 18th when the original version of 13.4 comes out , will i be able to upgrade the beta into the original without any issues and successfully run it without bugs . i 'm asking this because when i upgraded 12.4 to 12.10 it had a lot of glitches to it . will the 13.4 run the same after upgrading as if i was to install the it directly as it is ? **upgrade**, beta
d_q^-	can not upgrade from 11.10 for my samsung x05 laptop installed ubuntu 11.10 desktop , my samsung x05 laptop work perfectly now . but while what to upgrade to 12.4 lts or 13 , it tell me that my features not supported by the cpu , and can not install . does anything can do so as to upgrade ? thanks . **upgrade**, samsung, cpu
d_q^-	unsuccessful upgrade from 11.10 to 12.4 lts i had upgrade from ubuntu 11.10 , that runs alongside windows7 , to 12.4 lts in a netbook samsung nc110 . after upgrade ubuntu doesnt run . after log in all the screen is in blue . what can i do ? thanks for the help . 11.10, 12.04, **upgrade**, laptop, samsung
d_q^-	how can i upgrade from ubuntu 12.4 to 12.10 ? i want to upgrad from ubuntu 12.4 to 12.10 ? ? i tried from this suit http : //releases.ubuntu.com/12.10/ but it does n't work.. please help . upgrade
d_q^+	how to upgrade ubuntu from 12.10 to 13.4 without lossing anything ? well title says all most everything . i am using ubuntu 12.10 . installed using windows installer . is there anyway to upgrade it ? any how much data i will lose ? i have installed some softwares and there is some docs . **upgrade**
d_q^-	upgrading from netbeans 7.3 to netbeans 8.0 on ubuntu 13,10 is there a better way to upgrade netbeans without having to download the whole new version again and installing it ? am running ubuntu 13.10 , i want to upgrade my netbeans installation from 7.3 to 8.0 the only option i get is to download netbeans 8.0 and install it . so should i uninstall 7.3 first then in download and install 8.0 ? is there a better way of upgrading netbeans on ubuntu ? thanks in advance for your help 13.10, **upgrade**, netbeans
d_q^+	acer touch pad not working i have upgraded my ubuntu 12.4 to 13.4 , everything is fine but my touch pad is not working . why touch pad is not accepting clicks ? touchpad
d_q^+	problem upgrading from12.10 to 13.04 i recently tried to upgrade my system to 13.4 , but in the middle of the the upgrade , system crashed . when i try to restart it goes to a slow and sluggish mode . please help . 13.04
d_q^-	how can i upgrade to unity 6.8 without upgrading to 12.10 ?

i am running 12.04 and i want to upgrade just unity without upgrading the version . is this possible ? if yes , how can i do it ? ?
12.04, unity, 12.10

---

d_q^-

no desktop after upgrade from ubuntu 12.10 to ubuntu 13.4
after i upgraded to ubuntu 13.4 at first i could n't log in . i got the error " could not update ice authority " . afther deleteing the file " ice authority " i could n't see anything on the screen . also my " user " from the earlier install , is missing too , but i can see the files in the home/olduser directory . can any one help me out with this , i do n't want to format as i have a lot of files and contacts and other data saved in the applications .
13.04

---

d_q^-

system problem ! ubuntu 12.4
when i was upgrading to ubuntu 12.4 i had to leave and accidentally turned off the laptop in half update . when lit it not entered the system , it put the screen in black and looked just the mouse . i had to install ubuntu 12.4 from a dvd drive apart , when i installed told me that was already installed , it saids that if you wanted to install ubuntu 12.4 with ubuntu 12.4 and i did it , but now i have two ubuntu 12.4 and also can not find my files . what i can do ?
system

---

d_q^-

ubuntu 13.10 +amd 13.4 beta drivers . = black screen
hoping for some troubleshoot . fresh install of ubuntu 13.10 . i get into gui for the first time , and run sudo apt-get update sudo apt-get upgrade then i follow this guide : http : //wiki.cchtml.com/index.php/ubuntu_saucy_installation_guide (replacing the 13.12 driver with the beta ones) . (i did the auto install stuff that comes with the beta driver) . unfortunately on reboot , i am getting a black screen . the first black screen brought me to a place with a cursor that has an " x " . switching to ttys is available , but when i click ctrl alt f7 to get into my gui = nada . this is terrible . hopefully someone can help me out . thanks
13.10, ati, video-driver, beta

---

d_q^-

how to upgrade to the unstable ubuntu builds
is there a way to upgrade my ubuntu 12.04 install to trusty dev without upgrading via .iso because it keeps erasing my programs and my desktop environment that i 'm working on .
12.04, **upgrade**

---

d_q^-

is it recommanded to install fresh ubuntu (desktop) instead of upgrade ?
is upgrade from ubuntu old version to latest (lts) version takes long time and risky if power failure or connection fail ? many users facing problem after upgrading ubuntu . so , if upgrade takes long time and besides may be problematic , then is it should be recommended or preferable to avoid upgrade , first backup data and then install fresh ubuntu which does n't take more time and no risk or we can say that right way to establish and use latest ubuntu without any doubt ?
upgrade

---

d_q^-

ubuntu 14.4 is not supporting dell inspiron15-3537 vga card
i upgrade the my laptop ubuntu 13.4. in to 13.10 after that vga controller is not working . then i did a new clean install with the latest version of ubuntu (14.04) but the problem is still there . but it 's working finely with 13.4. but i cant update software repositories . pls help me to fix this bug .
14.04

---

d_q^-

ubuntu 13.4 upgrade download website ubuntu
i have ubuntu 13.4 i would like to upgrade this where can i find a patch ? google has nothing , can not find anything on your website . greetings , fabrizio
upgrade

---

d_q^-

how to upgrade ubuntu 12.04 to 12.10 final not beta without a cd
i tried doing this before from the software updater and it upgraded me to 12.10 and then i upgraded to 13.04 and then when i tried to upgrade to 13.10 it said that it ca n't because i 'm using a daily build so how do i avoid upgrading to a daily build from ubuntu 12.04 as i assume that the 12.10 was a daily build install too . also if i try to upgrade via cd/dvd it hangs when it asks me if i want to install updates and or proprietary software .
12.04, 13.04, 12.10, 13.10, **upgrade**

---

d_q^-

how do i disable the upgrade to new version warning ?
after i allowed ubuntu on my notebook to upgrade to 13.4 , it lost wi-fi connection . network menu does not show wireless access points , but according to syslog it establishes connection . also , the wireless router provides ip address to the host . i do n't want to spend time on debugging and solving the problem that most probably will reappear with new update , so i decided to reinstall 12.10 version . it recognized broadcom wi-fi modem right after the boot , and everything went great , until today i powered it on and suddenly found that it is running 13.4 and all my problems are back ! is it possible to instruct it to stay with 12.10 until i decide to upgrade ?
12.10, **upgrade**

---

d_q^-

why i should remove proprietary softwares and packages to upgrade my ubuntu
i have tried to upgrade my ubuntu twice now and have n't succeeded . the first was to upgrade from 13.04 to 13.10 . yesterday i tried to upgrade from ubuntu 13.10 to 14.04 using upgrade manager but without success . what i want to know is why it is so difficult ? is it really difficult to replace old packages with new ones ? if my wifi card is not supported how i do to upgrade if must remove the drivers ? i have filled a bug report here : the response i got : the version of perl as well a several perl modules installed on your system are from debian . please revert these packages to the version from the official ubuntu repository and try the upgrade again . why i must do that if i will remove every package i have installed so it 's not upgrade it 's fresh install + keeping data .
upgrade

---

d_q^+

how can i upgrade from 12.10 to 13.10 via terminal using usb rather than from internet ?

while upgrading from 12.10 to 13.10 i was unable to select the upgrade option as follow , the same happened while i was trying to upgrading from 12.10 to 13.04 booting from usb presently i have a dial up connection and it 's too slow to upgrade via internet . so how can i use terminal to upgrade ubuntu via iso image of 13.10 . i do n't want to make a fresh installation because i do n't want to loose the packages i have .
12.10, 13.10, **upgrade**, usb-drive, gnome-terminal

B Appendix

CNN	max pool, drop 0.2 L_q^{RQ2} loss	mean pool, drop 0.2 L_q^{RQ2} loss	max pool, drop 0.1 L_q^{RQ2} loss	max pool, drop 0.3 L_q^{RQ2} loss	trainable embeddings L_q^{RQ2} loss	randomized embeddings L_q^{RQ2} loss	max pool, drop 0.2, L_q^{RQ3} loss
max pool, drop 0.2 L_q^{RQ2} loss		0.0	0.99	0.98	0.83	0.94	0.98
mean pool, drop 0.2 L_q^{RQ2} loss			0.0	0.01	0.94	0.98	0.01
max pool, drop 0.1 L_q^{RQ2} loss				0.91	0.79	0.90	0.91
max pool, drop 0.3 L_q^{RQ2} loss					0.85	0.92	1.0
trainable embeddings L_q^{RQ2} loss						0.96	0.93
randomized embeddings L_q^{RQ2} loss							0.92
max pool, drop 0.2, L_q^{QR3} loss							

(a)

LSTM	max pool, drop 0.2 L_q^{RQ2} loss	mean pool, drop 0.2 L_q^{RQ2} loss	last pool, drop 0.2 L_q^{RQ2} loss	max pool, drop 0.1 L_q^{RQ2} loss	max pool, drop 0.3 L_q^{RQ2} loss	trainable embeddings L_q^{RQ2} loss	randomized embeddings L_q^{RQ2} loss	max pool, drop 0.2, L_q^{QR3} loss
max pool, drop 0.2 L_q^{RQ2} loss		0.84	0.0	1.0	0.50	0.28	0.32	1.0
mean pool, drop 0.2 L_q^{RQ2} loss			0.0	0.89	0.95	0.99	0.99	0.95
last pool, drop 0.2 L_q^{RQ2} loss				0.0	0.0	0.0	0.0	0.0
max pool, drop 0.1 L_q^{RQ2} loss					1.0	0.99	0.98	1.0
max pool, drop 0.3 L_q^{RQ2} loss						1.0	1.0	0.82
trainable embeddings L_q^{RQ2} loss							0.99	0.93
randomized embeddings L_q^{RQ2} loss								0.88
max pool, drop 0.2, L_q^{QR3} loss								

(b)

GRU	max pool, drop 0.2 L_q^{RQ2} loss	mean pool, drop 0.2 L_q^{RQ2} loss	last pool, drop 0.2 L_q^{RQ2} loss	max pool, drop 0.1 L_q^{RQ2} loss	max pool, drop 0.3 L_q^{RQ2} loss	trainable embeddings L_q^{RQ2} loss	randomized embeddings L_q^{RQ2} loss	max pool, drop 0.2, L_q^{QR3} loss
max pool, drop 0.2 L_q^{RQ2} loss		0.03	0.0	1.0	0.97	0.01	0.00	1.0
mean pool, drop 0.2 L_q^{RQ2} loss			0.0	0.09	0.29	0.65	0.63	0.07
last pool, drop 0.2 L_q^{RQ2} loss				0.0	0.0	0.0	0.0	0.0
max pool, drop 0.1 L_q^{RQ2} loss					0.99	0.98	0.98	1.0
max pool, drop 0.3 L_q^{RQ2} loss						1.0	1.0	0.99
trainable embeddings L_q^{RQ2} loss							0.75	0.87
randomized embeddings L_q^{RQ2} loss								0.92
max pool, drop 0.2, L_q^{QR3} loss								

(c)

Table 18. Modified p-values calculated by the randomized Tukey's HSD test.

Bi-LSTM	240-dim LSTM	400-dim, concat 1	240-dim, concat 1	240-dim, concat 0
240-dim LSTM		0.10	0.41	0.23
400-dim, concat 1			0.95	0.94
240-dim, concat 1				0.92
240-dim, concat 0				

Bi-GRU	280-dim GRU	400-dim, concat 1	240-dim, concat 1	240-dim, concat 0
280-dim GRU		0.22	0.45	0.74
400-dim, concat 1			0.03	0.55
240-dim, concat 1				0.36
240-dim, concat 0				

Table 19. Modified p-values calculated by the randomized Tukey's HSD test.