6.864, Fall 2017 - Neural Approaches to Question Retrieval

Codebase: https://github.com/shashank-srikant/6.864_term_project

Vadim Smolyakov*
CSAIL, MIT
vss@csail.mit.edu

Shashank Srikant*
CSAIL, MIT
shash@mit.edu

ABSTRACT

Content on the internet grows at an exponential rate. Given this growth, finding relevant information accurately becomes a critical task for the NLP community to address. More so, with this rapid growth, curating labeled datasets to build models for the wide variety of content available on the internet has become extremely time and resource intensive. In this work, we explore whether neural models are able to successfully model content similarity and retrieval tasks, and whether they can transfer knowledge from one domain, where supervised labels are available, to a domain with no available labels. Specifically, we explore content similarity in online discussion forums, where we explore the following questions - a. how effectively can neural approaches model question-answer retrieval tasks, which is, given a question and answer pair present on an online discussion forum, how effectively can neural approaches find similar pairs on that forum. b. Given a model of question-answer similarity in one online discussion forum, how effective are neural approaches in transferring that knowledge to a new, loosely related domain. In this work, we explore a baseline approach of modeling question similarity tasks on the popular online community *AskUbuntu.* We show how neural architectures like LSTMs and CNNs outperform traditional approaches in information retrieval. Additionally, and importantly, we explore the problem of transferring these models to detect question-answer similarity on *Android stack exchange*, a similar yet different online discussion community which discusses Android related problems. We show how neural domain adaptation techniques successfully beat baseline IR techniques and direct neural transfer techniques. We also discuss some limitations and challenges in using such architectures.

1. INTRODUCTION

The problem of text similarity, and specifically, similarity of short queries or answers found on the internet, have been central to the modern NLP community. With the explosion in content on the internet, a lack of robust tools to find similar content has the risk of creating further similar and redundant content, which only exacerbates the original problem.

Another relevant and pressing concern which such an explosion and variety of content has introduced is the increased cost of building predictive NLP, NLU models which cater to such content. The variety in content requires building a model from scratch, irrespective of how closely related the content may have been to a previously built model. For instance, if we were to model the reviews written for movies, we would have to reinvest effort and time in modeling reviews written for another domain, say, food or hotels. In spite there being conceptual, semantic similarities between the tasks of reviewing movies and reviewing food, models have to be created anew. And each such modeling exercise demands a large repository of curated, preferably labeled data, which most often is not feasible to put together. The pressing challenge such a variety of content has created is to be able to learn with minimal supervision, and from loosely related datasets.

In this work, we investigate two problems - one, to model question-similarity tasks using state of the art techniques in neural modeling and two, transfer those models to a loosely related, yet equally rich domain under the constraint of having no supervised information on that new domain. Specifically, we explore the problem of finding similar question-answer pairs on the popular online discussion *AskUbuntu*[1]. This discussion forum focuses on troubleshooting queries on Ubuntu, the popular open-source operating system. In addition to learning models to find relevant question-answer pairs on this forum, we also use this domain to transfer knowledge onto modeling question-answer similarity in *Android stack exchange*[2]. This is an online community, similar in setup and structure to *AskUbuntu*, wherein short queries and corresponding answers on Android-related troubleshooting are present.

In this work, we investigate how state of the art neural architectures for NLP applications perform on the task of question similarity. Traditionally, text similarity tasks were solved using information retrieval techniques like building an indexer and applying ABC and ABC, which consider XYZ. We study how more recent techniques like RNNs and CNNs, which have shown to successfully model text on tasks like XYZ, model this particular task. For the domain adaptation task of learning question similarity in Android stack exchange, we explore adversarial neural techniques, and Doc2Vec [?] in learning from the *Ask Ubuntu* dataset.

The study demonstrates the relevance of neural techniques in a critical NLP task like question-answer retrieval. We demonstrate how using neural approaches, one can outper-

*Author order decided by tossing a fair coin.

[1] https://askubuntu.com

[2] https://android.stackexchange.com

Scenario	Setting	Nature of Data	Learning Paradigm	Main Concepts
$\mathcal{D}_S = \mathcal{D}_T$, $T_S = T_T$	Traditional Machine learning	Labelled data in source domain(s) and unlabeled data in target domain	Source and target domains are exactly the same	Learn models on training set and test on future unseen data
$\mathcal{D}_S \neq \mathcal{D}_T$, $T_S = T_T$	Transductive Transfer Learning	Labelled data in source domain(s) and unlabeled data from $\mathcal{P}(X_S) \neq \mathcal{P}(X_T)$	Single source domain adaptation	Learning common shared representation; instance weighing, parameter transfer
			Multi-source adaptation	Classifier combination; efficient combination of information from multiple sources; Feature representation
No conditions on $\mathcal{D}_S$, $\mathcal{D}_T$, but $T_S \neq T_T$	Inductive Transfer Learning	Unlabeled data in source domain(s) and labeled data in target domain	Self-taught learning	Extracts higher level representations from unlabeled auxiliary data to learn instance-to-label mapping with labeled target instances
		Labeled data is available in all domains	Multi-task learning	Simultaneously learns multiple tasks within (or across) domain(s) by exploiting the common feature subspace shared across the tasks
$\mathcal{D}_S \neq \mathcal{D}_T$ $T_S \neq T_T$	Kim et al. (2015)	Labeled data in source and target domains	Transfer learning with disparate label set	Disparate fine grained label sets across domains, however, same coarse grained labels set can be invoked across domains

form traditional information retrieval techniques while not having to invest heavily in engineering the right features to get to such a performance. Additionally, we show the successful transfer of domain knowledge from one deomain to another using state of the art neural transfer techniques.

This work is organized into the following sections - Section 2 discusses related work in this field. Sections 3, 4, 5 discuss the various techniques to model in-domain and domain adaptation tasks for the given problem. Section 6 discusses the experiment setup and the results. Section 7 discusses the results and concludes our work.

2. RELATED WORK

Table summarizes the work in this domain.

3. IN-DOMAIN QUESTION SIMILARITY

3.1 Type Changes and Special Characters

We have already seen several typeface changes in this sample. You can indicate italicized words or phrases in your text with the command `\textit`; emboldening with the command `\textbf` and typewriter-style (for instance, for computer code) with `\texttt`. But remember, you do not have to indicate typestyle changes when such changes are part of the *structural* elements of your article; for instance, the heading of this subsection will be in a sans serif[3] typeface, but that is handled by the document class file. Take care with the use of[4] the curly braces in typeface changes; they mark the beginning and end of the text that is to be in the different typeface.

Citations to articles [**?**; **?**; **?**; **?**], conference proceedings [**?**] or books [**?**; **?**] listed in the Bibliography section of your article will occur throughout the text of your article. You should use BibTeX to automatically produce this bibliography; you simply need to insert one of several citation commands with a key of the item cited in the proper location in the `.tex` file [**?**]. The key is a short reference you invent to uniquely identify each work; in this sample document, the key is the first author's surname and a word from the title. This identifying key is included with each item in the `.bib` file for your article.

The details of the construction of the `.bib` file are beyond the scope of this sample document, but more information can be found in the *Author's Guide*, and exhaustive details in the *LaTeX User's Guide*[1].

So far, this article has shown only the plainest form of the citation command, using `\cite`.

4. DOMAIN ADAPTATION

5. OTHER TECHNIQUES FOR DOMAIN ADAPTATION

6. EXPERIMENTS

7. DISCUSSION

8. ACKNOWLEDGEMENTS

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to

[3] A third footnote, here. Let's make this a rather short one to see how it looks.

[4] A fourth, and last, footnote.

thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the **.cls** and **.tex** files that it describes.

9. REFERENCES

[1] G. Singh, S. Srikant, and V. Aggarwal. Question independent grading using machine learning: The case of computer program grading. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 263–272. ACM, 2016.