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 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 similarity tasks i.e. 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 compare to a traditional approaches in information retrieval. Additionally, and importantly, we explore the problem of transfering 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 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, in spite of how closely related the content may be. For instance,

With recent advances in neural architectures for NLPapplications, it is relevant to investigate how they perform on such tasks. Traditionally, text similarity has been approached with standard information retrieval techniques, like Lucene and REF, which consider XYZ. It is relevant to investigate how more recent techniques like LSTMs and CNNs, which have shown to successfully model text on tasks like XYZ, model this particular task.

In comparison to this, it is interesting to investigate whether using neural approaches, one can outperform them while not having to invest in engineering the right features to get to such a performance. In addition to

2. RELATED WORK

Typically, the body of a paper is organized into a hierarchical structure, with numbered or unnumbered headings for sections, subsections, sub-subsections, and even smaller sections. The command `\section` that precedes this paragraph is part of such a hierarchy.[1] LaTeX handles the numbering and placement of these headings for you, when you use the appropriate heading commands around the titles of the headings. If you want a sub-subsection or smaller part to be unnumbered in your output, simply append an asterisk to the command name. Examples of both numbered and unnumbered headings will appear throughout the balance of this sample document.

Because the entire article is contained in the **document** environment, you can indicate the start of a new paragraph with a blank line in your input file; that is why this sentence forms a separate paragraph.

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

*Author order decided by tossing a fair coin.

[1]This is the second footnote. It starts a series of three footnotes that add nothing informational, but just give an idea of how footnotes work and look. It is a wordy one, just so you see how a longish one plays out.

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[2] typeface, but that is handled by the document class file. Take care with the use of[3] 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*[**?**].

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

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

[3]A fourth, and last, footnote.