Enemble Learning in Quora Question Intent Matching: Using Parallel Naive Bayes, Word Embedding, and TF-IDF to Quantify Intent Differences

Nathan M. Brahmstadt
Department of Electrical and Computer Engineering
Oregon State University
brahmstn@oregonstate.edu

Jordan M. Crane
Department of Electrical and Computer Engineering
Oregon State University
cranejo@oregonstate.edu

Abstract

Our team set out to solve the problem set forth by the Kaggle Quora Question Pairs challenge; that is, to use machine learning and data mining techniques to identify Quora questions with similar intent. We aimed to build a model that given a pair of questions, could reliably identify whether the questions were duplicates or not. We pursued several avenues to find the best solution, but in the end we identified several meaningful features from classifiers that analyzed shared and different words. Data was filtered strategically to remove noise, fed into three classifiers, and combined with a logistic regression model. Through this ensemble approach, we achieved 0.4102 log-loss on our training data and 0.3552 log-loss on our testing data.

1 Approaches explored

In this section we detail the various approaches that we explored before settling on our final solution.

1.1 Preprocessing

Our preprocessing is primarily carried out by our file parser, which takes the raw CSV files and pares it down to something that is easier for our main classifier to work with. It performs several modifications on the data while it is being parsed.

1.1.1 Abbreviation substitution

One technique used by the parser is to look for common abbreviations and substitute in the expanded version. This is an extremely useful technique, since it provides the main classifier with more consistency across the data set.

1.1.2 Stop words

Another technique that we explored was the removal of stop words in the data. This approach had both positive and negative ramifications: while it did reduce the amount of fluff in the data, it also

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

removed some intent-altering words, such as when, where, and why. Ultimately, it resulted in lower log-loss so we decided to include it in our final parser.

1.1.3 Punctuation

In our initial implementation, the function written to strip punctuation did not do so properly, which was affecting our results significantly. However, when we changed it to strip all punctuation, we actually saw an increase in our log-loss. Further exploration revealed that this was caused by an interesting feature of the question pairs: in many cases, the last word of the question holds much greater weight with regard to the question's intent than the preceding words. By not removing the punctuation, we were unintentionally setting the last words apart since they were always considered by the classifier with the question mark included. In order to maintain this unexpected benefit while still taking adveantage of the gains provided by stripping punctuation, we decided to remove all punctuation, but then duplicate the last word, including one copy with the question mark and one without. This gave us the lowest log-loss of all three configurations.

1.1.4 Spell checking

A large amount of error in our classifier is the result of one-off misspellings in the dataset. To remedy this, we explored the idea of using a spell checker to correct the errors in the data. Several factors steered us away from this apporach. Firstly, we couldn't find a spell checking library that was fast enough to avoid unacceptable slowdown in our parser. Secondly, correcting all non-dictionary words to dictionary words can incur unintended side effects by changing intent-altering words that do not appear in common dictionaries, such as the names of foreign cities. Due to these considerations, we opted not to implement spell checking in our final solution.

1.2 Classifiers

1.2.1 Naive Bayes

We chose this classifier right off the bat due to our familiarity with it and how well it lends itself to the problem. However, previous implementations we had seen were calssifying documents independently of one another, not comparing two seperate texts as we are in this problem. In order to overcome this, we built an ensemble with two Naive Bayes classifiers: one classifier looked at the words which were common to the two questions, while the other looked at the set of words which differed. We found that these two in conjunction were able to make reasonable predictions on a large amount of the data, since most cases not covered by one were caught by the other. If either classifier was certain that a question was or was not a duplicate, it overrode the more uncertain classifier.

1.2.2 Word embedding

The other piece of our ensemble's top layer is Google's word2vec library implementing a pre-trained Google News corpus of 3 million 300-dimensional English word vectors. Word embedding is an interesting approach to natural language processing which represents words as vectors in a high dimensional space, such that words with similar usage in context will be placed similarly in the space. To use this classifier, we took the average of the vector values composing each question and then took the difference between these two values to get an idea of how contextually similar the questions were. This was definitely the least helpful of the classifiers used in our ensemble, but it provided a significant reduction to our log-loss, and more insight into certain question pairs. However, since we are using a pre-trained model word embedding falls short when it comes to misspellings.

1.2.3 Support vector machine

The final piece of our ensemble is a support vector machine to tie all the other components together. We chose to use an SVM for its simplicity and ability to group the outputs of the previous classifiers with wide margin.

1.2.4 Convolutional neural network

We also explored using a convolutional neural network to take the place of the SVM, but found it to be overly complex for the task. Additionally, we worried that a more complex model would overfit, and were unsure of our ability to combat this in an ensemble.

1.2.5 Long short-term memory network

1.2.6 Logistic Regression

1.3 Features

1.3.1 Common words

1.3.2 Unique words

1.3.3 Total frequency-inverse document frequency

1.3.4 Word sentiment

2 Results

In this section we explore the final solution upon which we settled. We based the structure of our algorithm with the following idea in mind: generate as many meaningful features as possible from the data, and use them to train a logistic regression model.

2.0.1 Preprocessing

Before the data was input into classifiers, there were many decisions considered

2.0.2 Common Word and Differing Word Naive Bayes Models

2.0.3 Sentence Similarity using Word Embedding

2.0.4 Term Frequency - Inverse Document Frequency

2.0.5 Logistic Regression

3 Conclusion

Our goal was to create meaningful classifiers exploring the concept of "differing and shared words" as identifiers of intent between two questions. The resulting ensemble approach unified three meaningful classifiers that we had discovered, and obtained a 0.35526 log-loss on the test data. This score is by no means ground-breaking, but we are satisfied by the result anyways. Our approach was able to perform above average in the competition, and we feel that the success it had, based on its relatively simple algorithm, exceeded our expectations.

3.1 Headings: second level

Second-level headings should be in 10-point type.

3.1.1 Headings: third level

Third-level headings should be in 10-point type.

Paragraphs There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.

4 Citations, figures, tables, references

These instructions apply to everyone.

4.1 Citations within the text

The natbib package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for natbib may be found at

```
http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf
```

Of note is the command \citet, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

> Hasselmo, et al. (1995) investigated…

If you wish to load the natbib package with options, you may add the following before loading the nips_2017 package:

```
\PassOptionsToPackage{options}{natbib}
```

If natbib clashes with another package you load, you can add the optional argument nonatbib when loading the style file:

```
\usepackage[nonatbib]{nips_2017}
```

As submission is double blind, refer to your own published work in the third person. That is, use "In the previous work of Jones et al. [4]," not "In our previous work [4]." If you cite your other papers that are not widely available (e.g., a journal paper under review), use anonymous author names in the citation, e.g., an author of the form "A. Anonymous."

4.2 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).

Note that footnotes are properly typeset *after* punctuation marks.[2]

4.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction. The figure number and caption always appear after the figure. Place one line space before the figure caption and one line space after the figure. The figure caption should be lower case (except for first word and proper nouns); figures are numbered consecutively.

You may use color figures. However, it is best for the figure captions and the paper body to be legible if the paper is printed in either black/white or in color.

4.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 1.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

[1] Sample of the first footnote.

[2] As in this example.

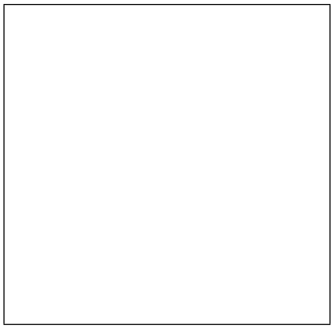

Figure 1: Sample figure caption.

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

`https://www.ctan.org/pkg/booktabs`

This package was used to typeset Table 1.

5 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

6 Preparing PDF files

Please prepare submission files with paper size "US Letter," and not, for example, "A4."

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You should directly generate PDF files using `pdflatex`.
- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- The IEEE has recommendations for generating PDF files whose fonts are also acceptable for NIPS. Please see `http://www.emfield.org/icuwb2010/downloads/IEEE-PDF-SpecV32.pdf`
- `xfig` "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
- The `\bbold` package almost always uses bitmap fonts. You should use the equivalent AMS Fonts:

  ```
  \usepackage{amsfonts}
  ```

  followed by, e.g., `\mathbb{R}`, `\mathbb{N}`, or `\mathbb{C}` for $\mathbb{R}$, $\mathbb{N}$ or $\mathbb{C}$. You can also use the following workaround for reals, natural and complex:

```
\newcommand{\RR}{I\!\!R} %real numbers
\newcommand{\Nat}{I\!\!N} %natural numbers
\newcommand{\CC}{I\!\!\!\!C} %complex numbers
```

Note that `amsfonts` is automatically loaded by the `amssymb` package.

If your file contains type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

6.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the `graphicx` package. Always specify the figure width as a multiple of the line width as in the example below:

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

See Section 4.4 in the graphics bundle documentation (`http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command when necessary.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

Appendix: Contribution Levels

Nathan

Jordan