Quora Question Pairs

The problem

When are two questions the same question? This is a problem faced by Quora, which has incoming questions from users that, to avoid duplication, have to be matched against questions that have already been posted.

A good place to start is that two questions are the same when they have the same words. However, it can (infrequently) happen that two sentences that have the same words mean different things, for example if the order is different. It can also (much more frequently) happen that two sentences that have completely different words mean the same thing. For example:

- Are autodidacts smarter than other people?
- How intelligent do you have to be to teach yourself hard subjects?

These two questions do not share words but are, generally, the same question.

Beyond comparing words, one can use word2vec vectors to try to capture and compare aspects of words. For example, word2vec might help you identify that a "car" is similar to an "auto" and to a "vehicle". That's useful if the problem you're working with is sufficiently narrowly defined.

This study is largely concerned with exploring distance measurements based on word2vec vectors as a proxy for meaning.

Preprocessing

Quora question pair study - preprocessing

quora_cleanup_1:
https://github.com/philopaszoon/capstone1/blob/master/qfin/quora_cleanup_1.ipynb

- using tabs instead of commas to separate columns
- cleaning up the IDs
- assigning group IDs to exact duplicates and metagroup IDs to questions that are scored as matches
- assigning keys to lines (linekey) to make it easy to merge data downstream
- dropping questions that have very few words after stop words are removed
- removing questions with non-english characters
- saving the data to a mysql database

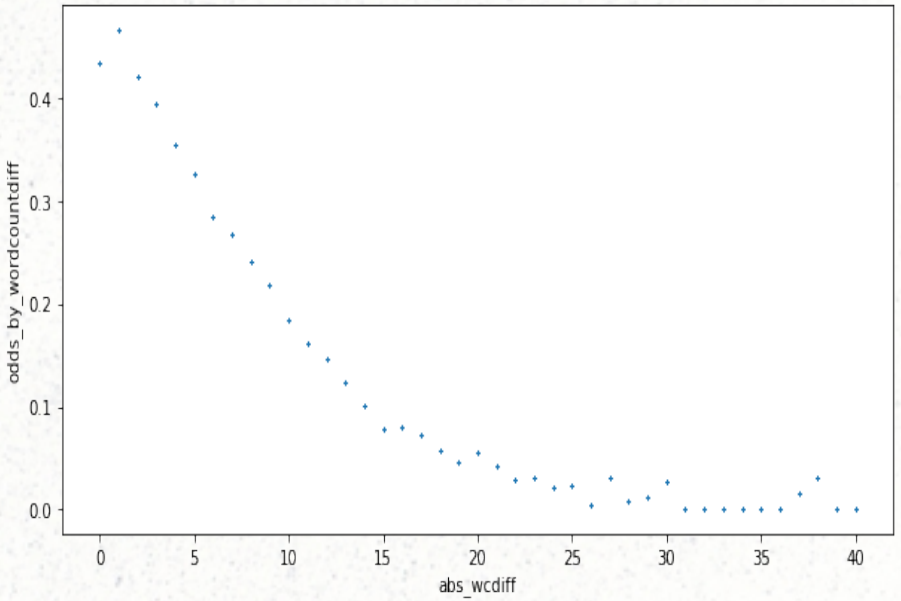
quora_contractions_3:
https://github.com/philopaszoon/capstone1/blob/master/qfin/quora_contractions_3.ipynb

- expanding contractions [he's -> he is, won't -> will not, etc]
- note that possessive contractions and contractions in named entities (which are mostly possessive as well) are not expanded

Adding Features – count features and quoted strings

quora_count_features_2: https://github.com/philopaszoon/capstone1/blob/master/qfin/quora_count_features_2.ipynb

- counting words and characters
- exploring the odds of a match by char and word counts and differences in word counts. As word count differences increase, the odds of a match shrink (right)



This is what one would expect to see if it's generally true that people try to be succinct. That is, similar concepts should require roughly a similar number of words to express.

quora_quoted_strings_4: https://github.com/philopaszoon/capstone1/blob/master/qfin/quora_quoted_strings_4.ipynb

- identifying sentences in which there is a quoted string.

Adding Features - first words(1)

quora_first_words_5:
https://github.com/philopaszoon/capstone1/blob/master/qfin/quora_first_words_5.ipynb

- In many languages, the first word of a question is often a marker of what kind of question it is. For example the question, "Who was your first grade teacher?", is obviously a question about a person. It's also an example of a placeholder question, where placeholders like Who, Where, Why, What,and When take the grammatical place of the object of the verb in the sentence. This can be better seen by rewriting the question in a declarative form, "Your first grade teacher [was] Who". Since a question about a person is unlikely to match a question about a place, simply recording what the first words are can be useful.

- First words can also be helpful in distinguishing yes/no questions from placeholder questions. Typically, questions that begin with a "to be" verb are yes/no questions. ex: "Are we there yet?". This class also includes words like Do, Would, Have, and others.

Adding Features - first words(2)

More than first words are being recorded in this notebook. There are also parts of speech related to first words and a group of columns to describe the “head” of the first word. A “head” is a root dependency of a given word. For example, in the sentence, “Who is the fastest runner on earth?”, the question word (Who) has the main verb (is) as its head word, which also happens to be the root of the sentence. Together with the first word, in some cases this can give a very low resolution snapshot of the meaning of the sentence as a whole.

ORTH	ORTH	LEMMA	TAG	DEP	POS	HEAD	HEAD_ POS	TREE
Who	Who	who	WP	attr	NOUN	is	VERB	[Who]
is	is	be	VBZ	ROOT	VERB	is	VERB	[Who, is, the, fastest, runner, on, earth, ?]
the	the	the	DT	det	DET	runner	NOUN	[the]
fastest	fastest	fast	JJS	amod	ADJ	runner	NOUN	[fastest]
runner	runner	runner	NN	attr	NOUN	is	VERB	[the, fastest, runner, on, earth]
on	on	on	IN	prep	ADP	runner	NOUN	[on, earth]
earth	earth	earth	NN	pobj	NOUN	on	ADP	[earth]

Word2vec (1)

What is word2vec

Word2vec evaluates the tendency of words to co-occur within a particular window (say, within 5 words). Instead of producing a prediction, the algorithm instead returns the weights that would be used to arrive at that prediction. Each word, then, is represented by a list of weights, each indicating the words participation in some aspect of meaning or sentence structure that would contribute to its odds of appearing in that place. For example, word2vec vectors can capture the difference between verbs and nouns, the difference between male and female and the difference between positive and negative. It is, for this reason, quite remarkable, especially considering the relatively simple way the result is achieved.

Vectors can be trained to any desired length and it's likely that different lengths are optimal for different problems. There seems to be a preference for vectors of between 300 and 400 weights. Performance and memory are certainly concerns with these, but it's likely that most problems simply don't benefit from using vectors much longer than this.

Word2vec (2)

Preparing word2vec models

I used three different kinds of word2vec models in this project.

1) a pre-trained model from Google
2) a pre-trained model from Spacy
3) a group of models trained on this quora data

Of these, the pre-trained models have the advantage of having been trained on very large volumes of data.

However, they do not have many of the named entities that are present in this data. I also wanted to explore word2vec models that included parts-of-speech and other information related to sentence structure.

Distance Measurements (1)

Distance Measurements

I explored two different kinds of distance measurement, Time Warped Distance and Cosine Similarity. These will discussed further in the next few slides.

Distance Measurements - Dynamic Time Warping (1)

Time Warped Distance measurement: (https://en.wikipedia.org/wiki/Dynamic_time_warping)

This measurement takes the idea of euclidian distance a little further by seeking the shortest distance between each word in one sentence and any word in the other sentence being considered. In a given pair of sentences [a b c] and [x y z], it can happen that point "a" in the first vector is closer to all the points "x", "y" and "z" in the other vector than either "b" or "c" is. This measurements permits that, taking the shortest available distance even if some points get left out or get used multiple times. This quality makes it good at getting the minimum possible distances between sentences. It's also good at finding similarity between sentences that are similar but misaligned.

Observe that euclidean distance does provide the base measurement between words themselves. That is, the timeseries being considered here is not [x y z] but words in a sentence, so it's the explicitly the distance between words, not letters or any other abtraction, that make up the unit elements of the calculation.

There is another version of the function in the file that can use either correlation or cosine similarity as a base measurement instead. Both versions are used, as I found that the version that uses euclidean distance is better at finding differences between sentences that are closely matched, while the one that uses correlation is better at finding differences between sentences that are a little further apart to begin with (see the named entity movement notebook).

Distance Measurements - Dynamic Time Warping (2)

```
def viDTWDistance(m1, m2):
    DTW={}
    for i in range(len(m1)): DTW[(i, -1)] = float('inf')
        for i in range(len(m2)): DTW[(-1, i)] = float('inf')
            DTW[(-1, -1)] = 0

            for i in range(len(m1)):
                for j in range(len(m2)):
                    dist= (distance.euclidean(m1[i],m2[j]))**2
                    DTW[(i, j)] = dist + min(DTW[(i-1, j)],DTW[(i, j-1)], DTW[(i-1, j-1)])

                    longer = len(m1)
                    if(len(m2) > longer): longer = len(m2)
                    rval =  (sqrt(DTW[len(m1)-1, len(m2)-1]))/longer
    return rval
```

Distance Measurements - Cosine Similarity (1)

This is the most commonly used method of measuring distance between vectors. The basic principle underlying this measurement is that a vector, in any number of dimensions, can always be thought of as having a particular "direction". While this is easy to visualize in two and three dimensions, the analogy remains valid in any number of dimensions. But if each vector has an assignable direction, it must also be true that there is a measurable angle between them.

So, how do we measure the "angle" between sentences? The first thing to note is that sentences don't have their own vectors. Word2vec gives us word vectors to play with, but there are relatively few domains in which whole sentences are used repatitively such that it would make sense to calculate vectors for them (such domains do exist, but quora sentence pairs is certainly not one of them).

Instead, we have to create sentence vectors by combining word vectors. Generally, this is done by simply averaging the word vectors from the words in each sentences down to a single vector for each sentence. As noted in the introduction, this process blurs sentences and loses information.

It has the advantage, however, of being fast.

Named Entities

quora_named_entity_counts_cm:
https://github.com/philopaszoon/capstone1/blob/master/qfin/quora_named_entity_counts_cm.ipynb

The nature of language is such that small details can have a significant impact on the meaning of a sentence. In this notebook deals with one class of those details, named entities. The objective is to pay special attention to the difference between sentences specifically where those differences are due to unmatched named entities.

This turns out to be a good use of the Dynamic Time Warp Distance measure, in part because the rotation (a term that will make sense after you look at the notebook) is almost completly obscured by the averaging which takes place as part of the cosine similarity measurement and in part because the TWDistance measurement is not overly sensative to word placement, so that as long as the same word gets the same rotation in both sentences, they'll be a match.

In simple terms, this notebook forces named entities to be move valuable than other words in the distance measurement between the sentences. If the sentences have the same named entities, it won't matter. But if they do have named entities, doing this helps to make it clear that the sentences are not the same.

Learning

quora_eval_999:
https://github.com/philopaszoon/capstone1/blob/master/qfin/quora_eval_999.ipynb

Finally, stitching all the columns together to pass through a random forest (I tried two of them, actually). It's pretty self explanatory. The one-hot encoded first_word data makes it look bigger and messier than it is.

The final result (accuracy = 0.77) was a bit less than I was hoping for, but not bad given a couple of factors:

1) I'm not using group IDs or estimated metagroup IDs at all. The questions are chronologically ordered and there is a slight tendency for larger ID values to be associated with the status of non-match because, as time went on, Quora did a better job of filtering out duplicate questions. This is discussed in detail on a Kaggle thread.
2) The original scoring is often suspect, ranging from obviously wrong to questionable.