

Results of Various Tests of K-Nearest-Neighbour (KNN) to Recognise a Paraphrased Statement

Lucian Green
RMIT, Melbourne, Australia

Abstract: In this paper, I tested a KNN algorithm that could recognise a paraphrased version of a statement entered in the Essay Helper GitHub Repository, which recognises reused statements while questioning the user to write a humanities-style essay.

Keywords: knn, paraphrasing, essay helper

1. Introduction to KNNs.

I found KNNs after querying "Prolog machine learning" on YouTube. K-Nearest-Neighbour was the successful candidate to select the best answer using machine learning while not sacrificing rule writing with the result. (They produce a list of results, i.e. 0 for exact match and increase with "greater distance" away from this.)

2. Tests of K-Nearest-Neighbour (KNN) to Recognise a Paraphrased Statement

In the tests, I wanted to test for success in finding that a statement contains a paraphrased statement.

First, Control: this test should return true if the statement is exactly the paraphrased statement:

```
?- Phrase="I'd like that very
much",downcase_and_split(Phrase,Key_Phase),get_string("",either,one,"",Key_Phase,String).
```

```
l: I'd like that very much
Attempt 1
[found,[i,d,like,that,very,much],[i]]
Success
```

The test results are below.

I tested for paraphrased statements in 6 word statements (surrounded by other words), where the replacement, addition or deletion of word(s) occurred at the start, middle or end of the 6 word paraphrased version of the statements.

Replacing 1 word with 1 word

```
Find "I'd like that very much." in "It goes unsaid that you'd like that very much my dear."
[found,[i,d,like,that,very,much],[you,d,like,that,very,much]]
```

```
Find "I'd like that very much." in "It goes unsaid that I'd like those very much my dear."
[found,[i,d,like,that,very,much],[i]]
```

```
Find "I'd like that very much." in "It goes unsaid that I'd like that very happily my dear."
[found,[i,d,like,that,very,much],[i]]
```

Replacing 1 word with 2 words

Find "I'd like that very much." in "It goes unsaid that you will like that very much my dear."
Failed

Find "I'd like that very much." in "It goes unsaid that I'd like those greatly much my dear."
[found,[i,d,like,that,very,much],[i]]

Find "I'd like that very much." in "It goes unsaid that I'd like that very happily most dear."
[found,[i,d,like,that,very,much],[i]]

Replacing 2 words with 1 word

Find "I'd like that very much." in "It goes unsaid that you like that very much my dear."
Failed

Find "I'd like that very much." in "It goes unsaid that I'd like those much my dear."
[found,[i,d,like,that,very,much],[i]]

Find "I'd like that very much." in "It goes unsaid that I'd like that very happily dear."
[found,[i,d,like,that,very,much],[i]]

Adding 1 word

Find "I'd like that very much." in "It goes unsaid that nevertheless I'd like that very much my dear."
[found,[i,d,like,that,very,much],[i]]

Find "I'd like that very much." in "It goes unsaid that I'd like that one very much my dear."
[found,[i,d,like,that,very,much],[i]]

Find "I'd like that very much." in "It goes unsaid that I'd like that very much love my dear."
[found,[i,d,like,that,very,much],[i]]

Adding 2 words

Find "I'd like that very much." in "It goes unsaid that in time I'd like that very much my dear."
[found,[i,d,like,that,very,much],[i]]

Find "I'd like that very much." in "It goes unsaid that I'd like that one there very much my dear."
[found,[i,d,like,that,very,much],[i]]

Find "I'd like that very much." in "It goes unsaid that I'd like that very much loved one my dear."
[found,[i,d,like,that,very,much],[i]]

Deleting 1 word

Find "I'd like that very much." in "It goes unsaid that I like that very much my dear."
[found,[i,d,like,that,very,much],[i]]

Find "I'd like that very much." in "It goes unsaid that I'd like very much my dear."
[found,[i,d,like,that,very,much],[i]]

Find "I'd like that very much." in "It goes unsaid that I'd like that much my dear."
[found,[i,d,like,that,very,much],[i]]

Deleting 2 words

Find "I'd like that very much." in "It goes unsaid that like that very much my dear."
Failed

Find "I'd like that very much." in "It goes unsaid that I'd like much my dear."
[found,[i,d,like,that,very,much],[i]]

Find "I'd like that very much." in "It goes unsaid that I'd like that my dear."
[found,[i,d,like,that,very,much],[i]]

Combinations

Replacing 1 word with 1 word at start
and adding 2 words in middle

Find "I'd like that very much." in "It goes unsaid that you'd like, indubitably so, that very
much my dear." Failed

Replacing 1 word with 1 word in middle
and adding 2 words at start

Find "I'd like that very much." in "It goes unsaid that in time I'd like those very much my
dear." [found,[i,d,like,that,very,much],[i]]
Adding 1 word in middle and

Deleting 2 words at start

Find "I'd like that very much." in "It goes unsaid that like that, timely, very much my dear."
Failed

Replacing 1 word with 2 words at start and
Deleting 1 word at end

Find "I'd like that very much." in "It goes unsaid that I definitely equally like that very my
dear." [found,[i,d,like,that,very,much],[i]]
Replacing 1 word with 2 words at end,
adding 1 word at start and
deleting 2 words in middle

Find "I'd like that very much." in "It goes unsaid that timelily I'd like definitely so my dear."
[found,[i,d,like,that,very,much],[i]]
Replacing 1 word with 2 words at start,

adding 1 word at middle and
deleting 2 words in end

Find "I'd like that very much." in "It goes unsaid that I definitely nonequally like that infinitesimally my dear." [found,[i,d,like,that,very,much],[i]]

3. Evaluation

The algorithm recognises the identical phrase and the phrase with only the first word changed, but fails if replacements or deletions involving multiple words are made at the start of the phrase. It can fail when trying to recognise the phrase with only the first word changed if there is another change.

4. Conclusion

The KNN algorithm that could recognise a paraphrased version of a statement entered in the Essay Helper GitHub Repository seemed mainly satisfactory. In the future, I would like to find an algorithm that can find key words and a correct paraphrasing of the sentence.

References

1. De Raedt, L. (2001). Relational Instance-Based Learning with Lists and Terms. [online] Available at: https://swish.swi-prolog.org/?code=https://raw.githubusercontent.com/samwalrus/machine_learning/master/classifiers/distances.pl [Accessed 26/10/2020].
2. Green, L. (2020). Essay Helper. [online] Available at: <https://github.com/luciangreen/Essay-Helper> [Accessed 26/10/2020].