KDnuggets

[Subscribe to KDnuggets News]

search KDnuggets Search

- Blog/News
- Opinions
- Tutorials
- Top stories
- Companies
- Courses
- Datasets
- Education
- Events (online)
- Jobs
- Software
- Webinars

Topics: Coronavirus | AI | Data Science | Deep Learning | Machine Learning | Python | R | Statistics

Comparison of the Text Distance Metrics

<= Previous post
Next post =>

Tags: Metrics, NLP, Text Analytics

There are many different approaches of how to compare two texts (strings of characters). Each has its own advantages and disadvantages and is good only for a range of specific use cases.

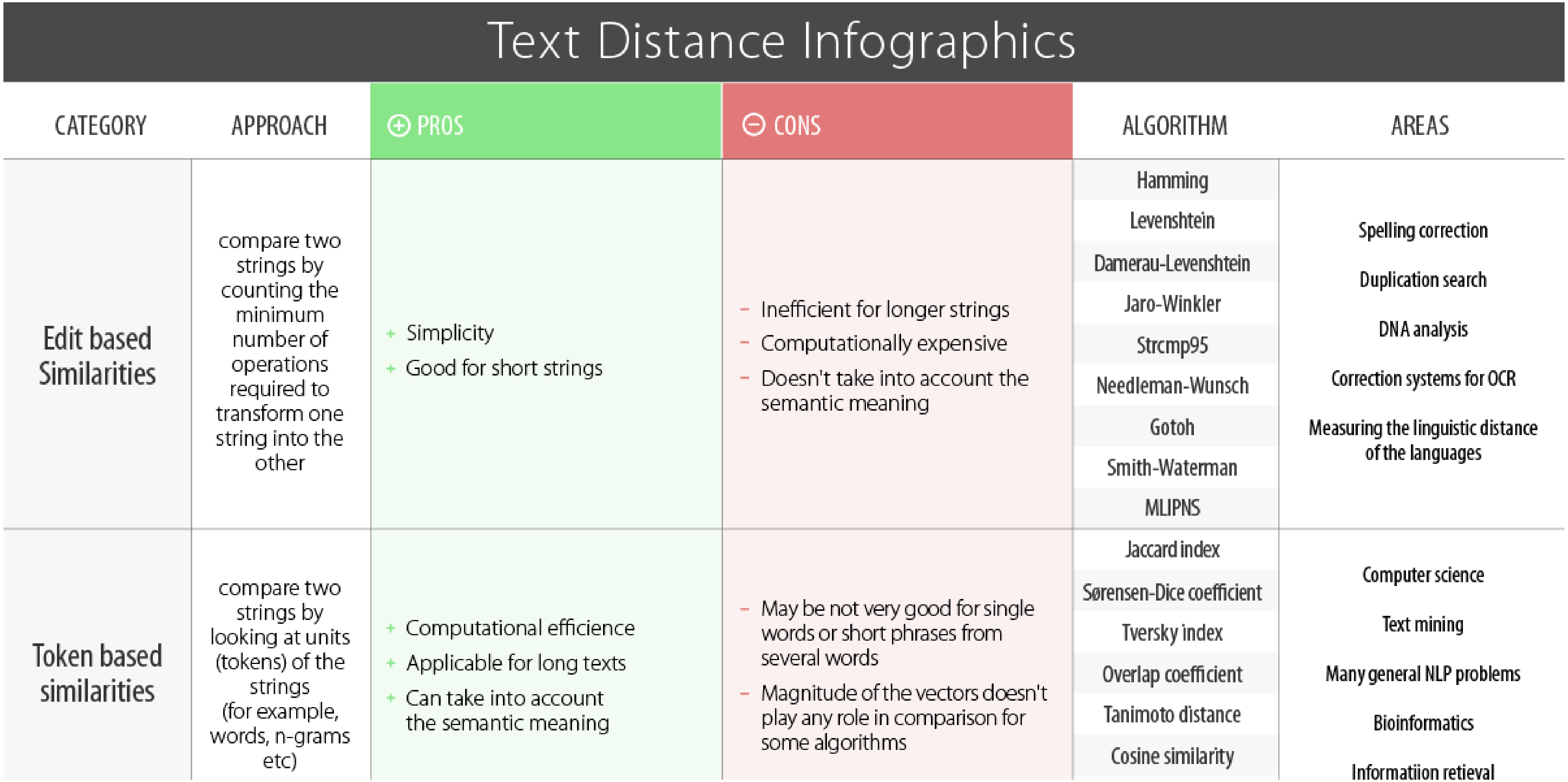
DOMINO
The future of enterprise data science is here
Learn how to accelerate research and manage data science at scale.
4.2
Discover what's new in Domino 4.2

comments

By ActiveWizards

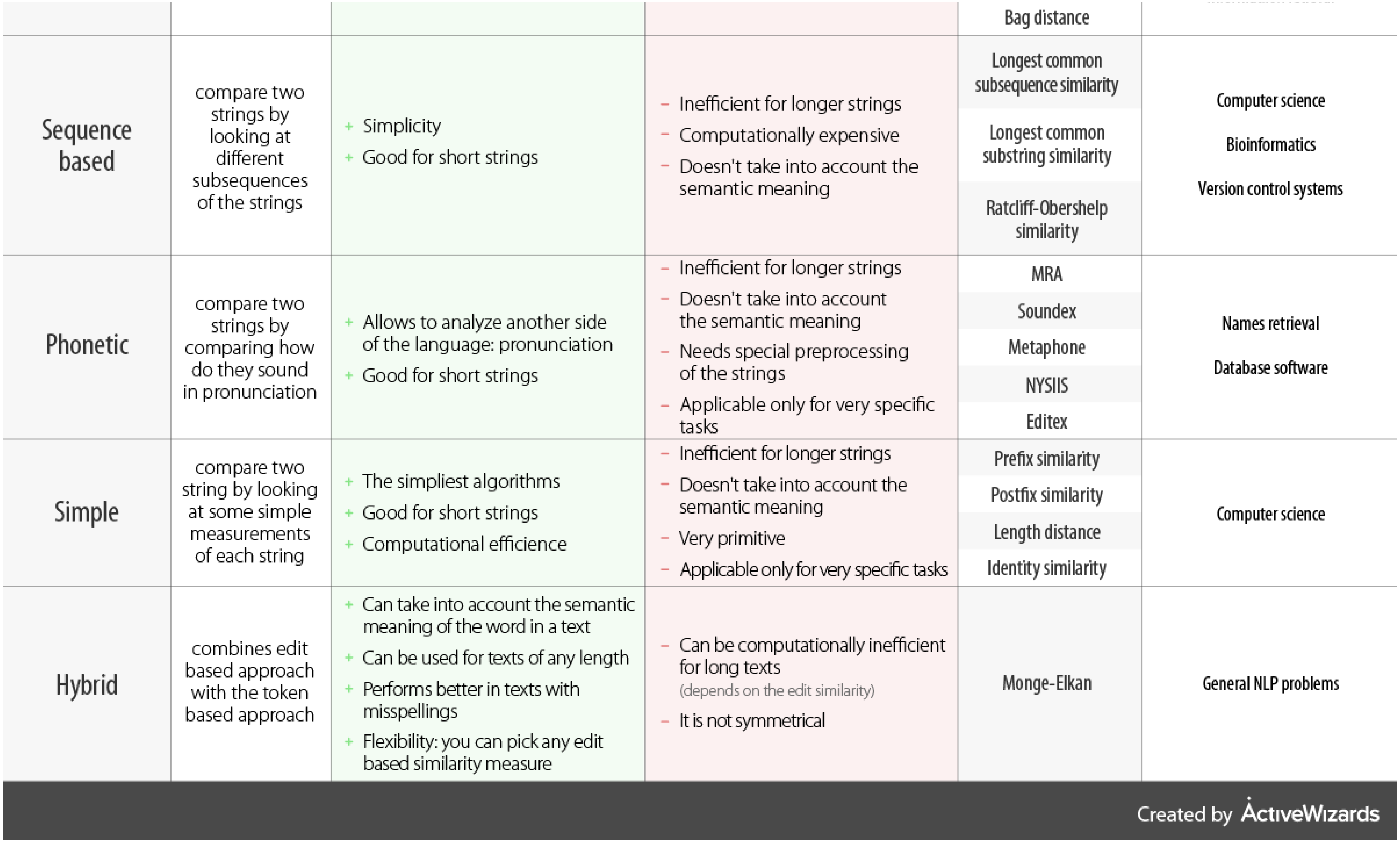
Header image

Measuring the similarity between texts is a common task in many applications. It is useful in classic NLP fields like search, as well as in such far from NLP areas as medicine and genetics. There are many different approaches of how to compare two texts (strings of characters). Each has its own advantages and disadvantages and is good only for a range of specific use cases. To help you better understand the differences between the approaches we have prepared the following infographic.



Text Distance Infographics

CATEGORY	APPROACH	⊕ PROS	⊖ CONS	ALGORITHM	AREAS
Edit based Similarities	compare two strings by counting the minimum number of operations required to transform one string into the other	+ Simplicity + Good for short strings	– Inefficient for longer strings – Computationally expensive – Doesn't take into account the semantic meaning	Hamming Levenshtein Damerau-Levenshtein Jaro-Winkler Strcmp95 Needleman-Wunsch Gotoh Smith-Waterman MLIPNS	Spelling correction Duplication search DNA analysis Correction systems for OCR Measuring the linguistic distance of the languages
Token based similarities	compare two strings by looking at units (tokens) of the strings (for example, words, n-grams etc)	+ Computational efficience + Applicable for long texts + Can take into account the semantic meaning	– May be not very good for single words or short phrases from several words – Magnitude of the vectors doesn't play any role in comparison for some algorithms	Jaccard index Sørensen-Dice coefficient Tversky index Overlap coefficient Tanimoto distance Cosine similarity	Computer science Text mining Many general NLP problems Bioinformatics Informatiion retieval



				Bag distance	
Sequence based	compare two strings by looking at different subsequences of the strings	+ Simplicity + Good for short strings	– Inefficient for longer strings – Computationally expensive – Doesn't take into account the semantic meaning	Longest common subsequence similarity Longest common substring similarity Ratcliff-Obershelp similarity	Computer science Bioinformatics Version control systems
Phonetic	compare two strings by comparing how do they sound in pronunciation	+ Allows to analyze another side of the language: pronunciation + Good for short strings	– Inefficient for longer strings – Doesn't take into account the semantic meaning – Needs special preprocessing of the strings – Applicable only for very specific tasks	MRA Soundex Metaphone NYSIIS Editex	Names retrieval Database software
Simple	compare two string by looking at some simple measurements of each string	+ The simpliest algorithms + Good for short strings + Computational efficience	– Inefficient for longer strings – Doesn't take into account the semantic meaning – Very primitive – Applicable only for very specific tasks	Prefix similarity Postfix similarity Length distance Identity similarity	Computer science
Hybrid	combines edit based approach with the token based approach	+ Can take into account the semantic meaning of the word in a text + Can be used for texts of any length + Performs better in texts with misspellings + Flexibility: you can pick any edit based similarity measure	– Can be computationally inefficient for long texts (depends on the edit similarity) – It is not symmetrical	Monge-Elkan	General NLP problems

Created by ActiveWizards

We highlight 6 large groups of text distance metrics: edit-based similarities, token-based similarities, sequence-based, phonetic, simple, and hybrid. The core features of each category are described in the infographic. Here, we just want to explain some nuances.

Edit based similarities are simple to understand. The more atomic operations you should perform to convert one string into another, the larger distance between them is observed. For example, the distance between words “hat” and “cat” is 1, and the distance between “map” and “cat” is 2. It is obvious that this approach is applicable only for words and short phrases but useless for longer texts. Also, this approach cannot take into account the semantic meaning of the words, because it compares only the characters. In the same time, we understand, that semantic distance between “lemon” and “apple” is less than between “lemon” and “moon” despite the fact that edit distance between “lemon” and “apple” may be greater than the distance between “lemon” and “moon”. That’s why edit based distances are used in the applications, where semantic meaning is not such important as similarity in writing. It is also worth to say that the most prominent edit based algorithm is the Levenshtein algorithm. Very often people think about edit based distances as about Levenshtein similarity.

Token-based similarities are a little bit more complex. Those analyze text as a set of tokens (words). This allows to take into account the semantic meaning of the words and to process large texts. Semantic meaning plays a role here because you can use word vector representations (word2vec) to describe each word in the text and then compare vectors. Also, using a bag of words approach and TF-IDF method allows comparing the semantic similarity between entire texts (although not between independent words). Token-based similarities are very widely used in different areas. Probably, it is the most well-known approach to work with texts. Nevertheless, it is not applicable to a range of use cases.

The next group of distance is sequence based distances. It is somewhat similar to edit based distances, but not completely the same. You may be guessing what is the difference between comparing strings on the basis of the longest common subsequence and the longest common substring. Longest common subsequence doesn’t take into account if there are some letters between characters from subsequence. For example, consider sets of letters “aebcdnlp” and “taybcrd”. The longest common subsequence between these words is “abcd”, while the longest common substring is only “bc”. The fields of application for these approaches are slightly different from edit based approaches, but the pros and cons are almost the same.

Phonetic algorithms form a separate group of methods for string comparison. In this case, it is not even string comparison but rather an audio comparison. These algorithms compare words based on how they are pronounced. It is hard to compare long texts in this way. The short sentences or phrases are the maximum threshold for these algorithms. Also, they cannot take into account the semantic meaning. Nevertheless, there are situations when these methods are indispensable.

The next several algorithms we want to mention are very simple to understand and use, so they form a group of methods which can be called as “Simple”. These methods can compare strings based on the similar prefixes or postfixes. Also, there are algorithms called “length distance” and “identity similarity”. The first ones compare strings by counting the number of characters in each of them and the second algorithms simply check if these strings are completely the same or not. As you can see, all these algorithms are very primitive and can be used only in very specific situations.

The last group we want to describe is hybrid algorithms. There is only one method: Monge-Elkan. It is a mix of edit based and token based distance. You can choose any edit based algorithm. The Monge-Elkan method compares each word in one text with each word in another text (so it is token-based), but when comparing words it uses some of the edit based methods (so it is edit-based at the same time). Then, distances between words are aggregated to derive a single value of the distance between the two texts. One important thing here is that this method is not symmetrical. This means that the result of comparison depends on what string you take as the first string and what as the second one. We describe this situation in the cons section of our infographic, but actually, it is not a problem for all applications. In other words, there are many use cases where symmetry is not important.

Conclusion

In this article, we have briefly described some interesting and important notes about different approaches to strings comparison. To understand and remember all these things better, please explore our infographics. There are no good or bad approaches, all of them appeared due to the need of using in particular cases. So, when you need to compare two strings, firstly think about what is the final result you expect to have, and then choose the right metric. Use our infographics as a cheat sheet!

ActiveWizards is a team of data scientists and engineers, focused exclusively on data projects (big data, data science, machine learning, data visualizations). Areas of core expertise include data science (research, machine learning algorithms, visualizations and engineering), data visualizations (d3.js, Tableau and other), big data engineering (Hadoop, Spark, Kafka, Cassandra, HBase, MongoDB and other), and data intensive web applications development (RESTful APIs, Flask, Django, Meteor).

Original. Reposted with permission.

Related:

- Comparison of Top 6 Python NLP Libraries
- Top 20 Python Libraries for Data Science in 2018
- Top 20 R Libraries for Data Science in 2018

Top Stories Past 30 Days

Most Popular

1. How Much Math do you need in Data Science?
2. Easy Speech-to-Text with Python
3. Learning by Forgetting: Deep Neural Networks and the Jennifer Aniston Neuron
4. Speed up your Numpy and Pandas with NumExpr Package
5. The Best NLP with Deep Learning Course is Free
6. If you had to start statistics all over again, where would you start?
7. Getting Started with TensorFlow 2

Most Shared

1. How Much Math do you need in Data Science?
2. A Complete guide to Google Colab for Deep Learning
3. 4 Free Math Courses to do and Level up your Data Science Skills
4. Easy Speech-to-Text with Python
5. Natural Language Processing with Python: The Free eBook
6. Uber's Ludwig is an Open Source Framework for Low-Code Machine Learning
7. Understanding Machine Learning: The Free eBook

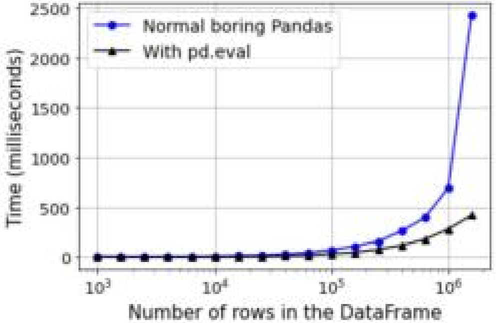
Latest News

- Some Things Uber Learned from Running Machine Learning ...
- A Complete Guide To Survival Analysis In Python, part 1
- 5th International Summer School 2020 on Resource-aware ...
- PyTorch for Deep Learning: The Free eBook
- Scope and Impact of AI in Agriculture
- Top Stories, Jun 29 – Jul 5: Speed up your Numpy ...

Top Stories Last Week

Most Popular

1. Speed up your Numpy and Pandas with NumExpr Package



2. How Much Math do you need in Data Science?
3. Getting Started with TensorFlow 2
4. Feature Engineering in SQL and Python: A Hybrid Approach
5. An Introduction to Statistical Learning: The Free eBook
6. Deploy Machine Learning Pipeline on AWS Fargate
7. Data Cleaning: The secret ingredient to the success of any Data Science Project

Most Shared

1. Deploy Machine Learning Pipeline on AWS Fargate
2. Getting Started with TensorFlow 2
3. Speed up your Numpy and Pandas with NumExpr Package
4. An Introduction to Statistical Learning: The Free eBook
5. Software engineering fundamentals for Data Scientists
6. Data Cleaning: The secret ingredient to the success of any Data Science Project
7. Data Scientists Have Developed a Faster Way to Reduce Pollution, Cut Greenhouse Gas Emissions

KDnuggets Home » News » 2019 » Jan » Tutorials, Overviews » Comparison of the Text Distance Metrics (19:n02)

Subscribe to KDnuggets News

X

What do you think?

6 Responses

👍 Upvote 😝 Funny 😍 Love

😯 Surprised 😤 Angry 😥 Sad

Comments **Community** 🔒 **Privacy Policy** **Login**

1

♡ **Recommend** Tweet Share **Sort by Best**

Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

<= Previous post
Next post =>