May 22, 2017 Report

Close Topic Clustering For Keyword Extraction:

The idea behind this method is since we are working on automatic keyword extraction of school children, any description of a concept presented on the slide, will mostly be about a single topic. The algorithmic model we present is based on that assumption.

Algorithm:

1) Repeat *N* times
    a. Choose one word from the list of words in the current document at random. Add the word and index to two lists.
    b. For the number of keywords requested,
        i. Create an empty list to store distances from the current word
        ii. For all the words
            1. If the index of the word is not in the added-indexes list,
                a. Calculate cosine distance between the current word and all the words in the document.
                b. Add the distance to the distance-list
            2. If the index of the word is in added-indexes list,
                a. Add some high value of the distance-list to act as a placeholder.
        iii. Find the minimum distance value and the index where it occurs
        iv. Add the index and distance to their respective lists.
        v. Change the current index to the minimum index from step (iii)
        vi. Add the word with the minimum distance from the current word to the list.
    c. Add all the words generated to a list
    d. Add the sum of distances stored at each repeat to a new list
2) Find the location where the sum of distances of all the keywords generated is the least.
3) Find the words added to the list from the location value in the step above.

Issues:

Why repeating? This is to ensure that all the initializations are considered and to overcome the slow start phase. In the example below, a good and a bad initialization case are shown.

Good Initialization

The Ramayana is a story of Lord Rama written by the Sage Valmiki. Lord Rama, the prince of Ayodhya, in order to help his father Dasaratha went to exile for fourteen years. His wife, Sita and his younger brother Lakshmana also went with him. He went through many difficulties in the forest. One day Ravana, the king of Lanka carried away Sita with him. Then, Lord Rama, with the help of Hanuman, defeated and killed Ravana; Sita, Rama and Lakshmana returned to Ayodhya after their exile.

Bad Initialization

The Ramayana is a story of Lord Rama written by the Sage Valmiki. Lord Rama, the prince of Ayodhya, in order to help his father Dasaratha went to exile for fourteen years. His wife, Sita and his younger brother Lakshmana also went with him. He went through many difficulties in the forest. One day Ravana, the king of Lanka carried away Sita with him. Then, Lord Rama, with the help of Hanuman, defeated and killed Ravana; Sita, Rama and Lakshmana returned to Ayodhya after their exile.

How do we know the difference between a good and bad initialization? It is simple. If we have a good initialization, our assumption that all the keywords belonging to the same topic will be closer holds. Therefore, when we sum up all the distances between the words, that ***total distance between the keywords will be least.*** On the other hand, if we start off with a word, say "difficulties", the relatively closest word would still be somewhat far. Therefore, the final sum of the distances would be more.

Capitalization

In the output results, we are currently getting a repetition of the same word due to capitalization of words in the document. Easy solution: Convert all the words in the text into lower case. We cannot. Proper nouns are always spelled with initial capital letters. Therefore, there exists a word vector for a word, say Mahabharata, but not for mahabharata. We are working on a remedy for this now.

Plural Forms

For example, if the document contains two ways, say male and males, though they might occur in different contexts, the distance between the singular and the plural form in the vector space is very small. To overcome this, we will stem all the words to their singular form, and we will partially work with them.

Missing Words

We, as a substitution, are using a single random vector for all the missing words. The issue with this strategy is quite inevitable here. For example, if there are four missing words in the document, and assuming that they are unrelated, when we calculate the cosine distance between them, the value shall be zero. In the end, when we add up all the distance between keywords, the set with these four missing words would be more, resulting in a lot of false positive results. Since this method is working fine till now, we have to find a remedy as soon as possible for this problem.

Definition – Example

We do not think this as an issue, but it sure is a problem that needs to be dealt in some way to reduce the intensity of it. Let us take a look at the example below.

> The Solar System consists of the Sun Moon and Planets. It also consists of comets, meteoroids and asteroids. The Sun is the largest member of the Solar System. In order of distance from the Sun, the planets are Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune and Pluto; the dwarf planet. The Sun is at the center of the Solar System and the planets, asteroids, comets and meteoroids revolve around it.

This document is an example where the model could easily fall into the trap of definition-example issue. If we set the number of keywords to 10, and if the initial word picked was something like Uranus, then all the words closest to that will be the other planets. Therefore, we are picking up a lot of examples that lend to the total concept, rather than the actual concept. Since the words picked up will be close, our termination condition that chooses the set of extraction keywords with the least intra-distance will also not help much.

Slow Start

This issue was resolved when we increased the number of repetitions.

More than one topic

This algorithm does word-topic selection in a different way. Assume a document has some time related words (fourteen, days, months) and definitive words (photosynthesis, chlorophyll, pigment). When the algorithm is initialized with the word photosynthesis, it first captures all the words belonging to that concept in the document. Once all these words are done, by some chance (as in closest far distance), it latches on to the time related word, which then creates a new sequence. This is how "clusters" are formed in the model.

Expansion Idea: Use some threshold. If the distance between the current word and closest far word is greater than some threshold, then we have to assume that all the words that belong to one topic were collected, and now it is time of create another sequence (or) "cluster" of keywords.