Article Duplicates

Brian Lynnerup Pedersen

Kongens Lyngby 2014

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Matematiktorvet, building 303B,
DK-2800 Kgs. Lyngby, Denmark
Phone +45 4525 3351
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

The goal of the thesis is to - snaps

Summary (Danish)

Målet for denne afhandling er at lave noget forskelligt

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring an B.Eng. in Informatics.
The project is equal to 20 ECTS points.

The thesis deals with the issue of finding articles that are duplicates in a large corpus of articles. This is done using various algorithms and is implemented in C#.

The thesis consists of ...

Lyngby, 09-June-2014

Brian Lynnerup Pedersen

Acknowledgements

I would like to thank my supervisors from DTU, Inge Li Gørtz and Philip Bille, for the help they provided to my project. Also I would like to thank my company Infomedia, for letting me do my project with them, my project leader Klaus Wenzel Jørgensen and Rene Madsen.

Contents

Chapter 1

Introduction

I have been working for Infomedia[1] since my third semester as an IT student at DTU (march 2012). Infomedia is in short a company that deals with news monitoring.

Infomedia is the result of a fusion between Berlings Avisdata and Polinfo in 2002, which means that Infomedia is partly owned by JP/Politikens Hus[2] and Berlingske Media[3]. It is a company with around 130 employees, of which a fair amount is student aides like myself. Infomedia has various departments, which includes an economy, sales, analysis and an IT department amongst others.
I am employed in the IT department as a student programmer.

Infomedia deals with news monitoring, which means that we have an inflow of articles from various newspapers, news sites, television and radio media, which we then monitor for content that is of interest to our clients. This can be a client that wishes to know when their firm is mentioned in the press or a product they are using, if that is being mentioned. We have also begun monitoring social media. Infomedia then sells various solutions to clients, for them to get this news monitoring.

[1] www.infomedia.dk
[2] www.jppol.dk
[3] www.berlingskemedia.dk

One of the things that we try to do in Infomedia, is that we want to present our clients with a fast overview of the articles in which terms[4], that trigger our news monitoring, appear. Many local newspapers are today owned by bigger media houses (like the owners of Infomedia) and as such, they will feature a lot of the articles that have also been printed in the "mother paper". This will make the same (or roughly the same[5]) article appear many times in news monitoring. In an effort to make the list of articles presented to the clients, easy to look at, and preventing a client having to read the "same" article many times, Infomedia has a wish to cluster article duplicates. Infomedia can then present the client with a list of articles and in that list have further sub lists that contains duplicates of the original aritcle[6].

Another issue, is the issue of copyrights and when the same article will appear in different media, but without content given from the writer of that article. An example that is often happening is that news telegrams from Reuters[7] or Ritzau[8] is published in a newspaper, but without the source indication. All news media are of course interested in knowing when their material is being published in competing media. This how ever can be tricky business, as official rules on the matter is incredible fuzzy.

I will in this thesis try and look into various ways of identifying article duplicates with in a test corpus[9] of articles. The long term goal for Infomedia is having this being implemented in the inflow of articles, and having a look back functionality so that we can group duplicates not just for one day, but for a longer period of time.

[4] A term is, in short, a word or a combination of words.

[5] Articles can be slightly edited in order to make them fit into the layout of the various papers.

[6] Or the longest article rather, as this will tend to contain the most information.

[7] www.reuters.com

[8] www.ritzau.dk

[9] A days worth of articles from 10/31/2013 - totalling 22.787 articles.

Chapter 2

General About Terms and Rules

blah about what words and terms is being used in the thesis

blah about copyright rules in Denmark...

Chapter 3

Algorithms in General

Considerations regarding pick of algorithms.

Bag of Words (Cosine), generates a vector from each document. Another document can then be matched with the list of documents, and their vectors are compared based on the dot product. The closer the dot product are to the value 1.0 the more similar are two documents. A score of 0.0 indicates that two words has nothing in common.

Problem: This string comparison is very sensitive to short articles, for instance "A man walks his dog in the park" and "A dog walks his man in the park" would result in returning a cosine value of 1.0, due to the term vector. This problem is very unlikely to yield false positives.

Longest Common Substring (LCS), compares documents in pairs. A general implementation would be to have list of documents and then compare a document to each of the documents in the list. The algorithm will then return the length of the longest common substring. This would probably needed to be modified to return a percentage match in stead (total string length / longest common substring found).

Problem: This algorithm is very prone to fail in cases where there have been made alterations to the article in question. A word change in the middle of

one of two otherwise identical articles will result in a 50 percent match. If the article in question have been obfuscated with many changed words, the LCS will be extremely short. This is a high risk problem, as article duplication will often involve changing words.

3.1 Optimizing Performance

3.1.1 Stop Words

Stop word removal would improve running time (performance) of both algorithms, and would pose little threat of causing either algorithm to fail. The exception to this could be very short articles (like breaking news articles), that only contains common words, like "Man walks away". Depending on the stop word list, this article could end up being *null*. We can safely (with respects to the previously addressed problem) remove stop words, as they don't provide any *semantic* value to the text.

3.1.1.1 Cosine

For the cosine algorithm removal of stop words would improve performance, by reducing the size of the *Magnitude Vector*.

3.1.1.2 LCS

In regards to the LCS algorithm, the performance would also be improved, as the algorithm wouldn't need to traverse as many *terms*. This would reduce the length of the longest common substring, but it would be unlikely that it would affect the outcome of the algorithm.

3.1.2 Stemming

Stemming would can improve performance of the cosine algorithm. This would not affect accuracy as stemming does not ruin the semantic content.

3.1.2.1 Cosine

Stemming improves the performance of the cosine algorithm as this will reduce the number of words in the *vector space*. As words would be reduced to their base form (Danish: *Grundform*), words used several times, but with different endings would be counted as the same word. When using weighed evaluation this would actually improve performance.

3.1.2.2 LCS

Stemmning would not improve the performance of the LCS algorithm. As this algorithm matches terms one by one, it would make no difference if the words are stemmed or not. For LCS it would actually decrease performance to apply stemming, as this operation would consume time while preparing the files, with no impact of the algorithms actual performance.

3.2 Implementation of Stop Words and Stemming

For the algorithms tested removing stop words would be an improvement (**TEST!!!**), and would be valuable to do before using either algorithm.

Stemming would should not be done when using the LCS algorithm as that has no performance enhancing impact on the LCS.

3.3 Semphore Tagging

Kig på semaphore tags? Samme artikel == samme tags?

Duplikat problematik, internet nyheder - historien opdateres løbende, hvornår bliver det et plagiat? Er samme artikel, men alligevel ikke...

Appendix A

Appendix

This appendix is full of stuff ... add more

Document from Infomedia about the general Vector Spaced Search.
Thesis document from Inge
Links to Wikipedia
The algorithm course book..