Identifying Online News Articles

By: Adam Standke
Mentor: Dr. Girard

I. Problem Statement

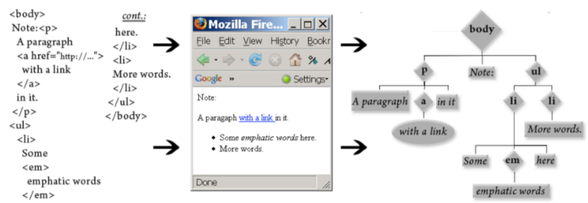
Does a news article exist within a web page?

II.Literature Background

Link-Target Identification: the structure of a hyperlink is analyzed to determine whether it points to relevant content or irrelevant content

CoreEX: is a heuristic based algorithm that extracts the main article from an online news website by analyzing the amount of text and number of links in every node in the Document Object Model tree(ie., DOM tree)

EXAMPLE OF AN HTML DOCUMENT, ITS SCREEN MANIFESTATION, AND ITS DOM TREE



COREEX'S HANDLING OF TERMINAL NODES

CoreEx handles terminal node's in one of three ways. If the terminal node is a text node (ie., a node that only contains text and no hyperlinks) CoreEx count's the total amount of words. If the terminal node is a link node (ie., a node that contains a hyperlink within it), CoreEX count's the link as one word of text. Otherwise, for all other types of terminal nodes, CoreEx does not record a value.

COREEX'S HANDLING OF NON-TERMINAL NODES

For a non-terminal node CoreEx iterates through the node's children and keeps track of the children whose text-to-hyperlink ratio is above a certain predetermined threshold. If this threshold is met, the node is scored based on the total amount of words and links contained by its children.

COREEX'S SCORING FUNCTION FOR NON-TERMINAL NODES

$$weight_{ratio} \times \frac{setTextCnt(N) - setLinkCnt(N)}{setTextCnt(N)} + weight_{text} \times \frac{setTextCnt(N)}{page_{text}}$$

CoreEx's scoring function has two components. The first term measures the ratio of the amount of text to the number of hyperlinks contained in the node. The second term captures the fraction of the total text in the web page that is contained in the node.

COREEX'S VARIABLES

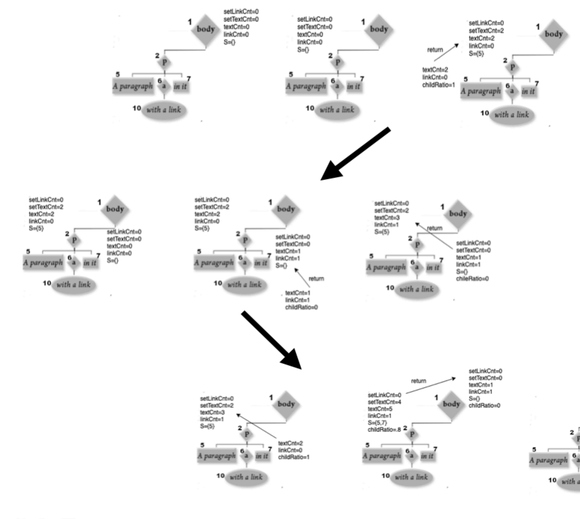
textCnt: holds the number of words contained in the node
linkCnt: holds the number of links in or below the node
S: represents the set of nodes that contain the main content
setTextCnt and **setLinkCnt** holds the sum of **textCnt** and **linkCnt** for nodes added to the set **S**
Threshold: determines whether a node's child should be added to the set **S**

Sources:
Ferreira, Rodolfo, et al. "Appling Link Target Identification and Content Extraction to Improve Web News Summarization." *Proceedings of the 2016 ACM Symposium on Document Engineering - DocEng 16*, Sept. 2016, pp. 197–200., doi:10.1145/2960811.2967158.

Prasad, Jyotika, and Andreas Paepcke. "Coreex." *Proceeding of the 17th ACM Conference on Information and Knowledge Mining - CIKM 08*, Oct. 2008, doi:10.1145/1458082.1458295.

Quinlan, John Ross. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 2006.

EXAMPLE OF COREEX'S TRAVERSAL OF DOM TREE



C4.5: is a machine learning algorithm that constructs a classification model in the form of a decision tree where each leaf indicates a particular class and each non-leaf represents a decision to be carried out on an attribute

$$gain(X) = info(T) - info_X(T) \text{ where,}$$

$$info(T) = -\sum_{j=1}^{k} \frac{freq(C_j, T)}{|T|} \times log_2(\frac{freq(C_j, T)}{|T|}) \text{ and}$$

$$info_X(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times info(T_i)$$

$gain(X)$ measures the information that is gained by partitioning a set of training cases by a given attribute

$info(T)$ calculates the average amount of information needed to identify the class of a training case

$info_X(T)$ represents the expected information after the set of training cases has been partitioned by an attribute

C_j represents a class, **T** represents a set of training cases, T_i represents a subset of training cases, and **X** represents an attribute

III. Research Objective

Assess the predictability of a news article existing within a web page

Limitation: 120 person-hours over 6 weeks

IV. Solution Description

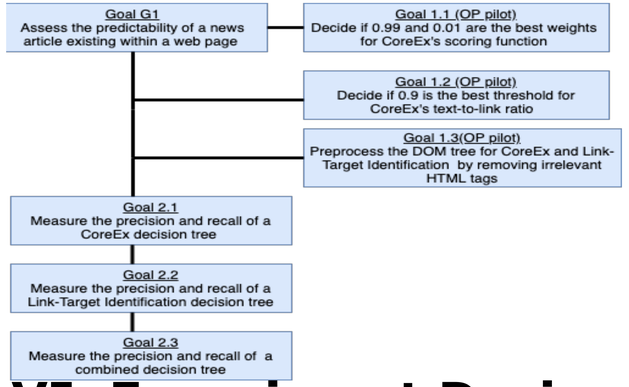
Using the C4.5 machine learning algorithm decision trees will be constructed from attributes extracted from Link-Target Identification and CoreEx

V. Hypotheses and Goal Tree

H_0 : Link-Target Identification will not perform better than CoreEx
H_1: Link-Target Identification will perform better than CoreEx

H_0 :Combining Link-Target Identification along with CoreEx will not produce the best performance

H_2: Combining Link-Target Identification along with CoreEx will produce the best performance



VI. Experiment Design

Factors	Values
Link-Target Id	{ does the root link contain a number/Id in the name of the link, does the root link contain a date, does the root link contain a reserved word, does the root link end with a forward slash mark, the length of the root link }
CoreEx	{ the main content node's HTML tag, the HTML tag with the highest frequency count in the set S, an integer that represents the score for the main content node }
Sample	Cross-validation will randomly split up 330 web pages into 10 groups of 33 web pages. Nine of the ten groups will be used to train the model and one group will be used to test the model. This process will be repeated for 20 trials.

	Decision tree built from CoreEx	**Decision tree built from Link-Target Id**	**Decision tree built from Link-Target Id and CoreEx**
297 training pages/33 test pages	X	X	X