Wikipedia Real-time Updates Recommendation System

Manoj Sirvi manoj.sirvi@research.iiit.ac.in
Zishan Kazi zishan.kazi@research.iiit.ac.in
Vikram Pudi vikram@iiit.ac.in

Abstract

With the rapid growth of Wikipedia, regular maintenance of its content requires corresponding growth in the laborious work of Wikipedians. We present a system that filters out most of the content and recommends reliable and worthy information that is a strong candidate for Wikipedia updation. Preliminary results show that our model achieves close to 98% sensitivity.

Keywords: Reliability, Web Retrieval, Feature Extraction, Feature Importance, Classification

1 Introduction

Real-time manual updates to Wikipedia pages is a labour-intensive task, especially for categories that require daily and real time updating, such as sports and politics. There is a need for an automatic real-time update recommendation system to help update Wikipedia with fresh, reliable content. We design and conceive that any such system would need to process the following three steps:

1. **Where to get:** Determine reliable sources that provide credible information.
2. **What to get:** Extract reliable and worthy sentences to add to Wikipedia by cross-verifying across sources.
3. **Where to add:** Determine where to add content in the Wiki page.

As a human, following the above steps might be easy but time-consuming. But for any machine, following these steps is even more complex. An MIT research group addressed the third problem of where to add content in Wikipedia pages (Matheson, 2020). In this paper, we tackle the first two problems and present an algorithm that completes the system to recommend reliable sources and worthy information for adding to Wikipedia.

To demonstrate the system's effectiveness, we tested it using the Wikipedia category for Hindi film actors like in (Pochampally and Karlapalem, 2017). This approach can be extended to other categories that require daily and real-time updating, such as Football players, Indian politicians, or American singers.

2 Problem

Given a Wiki page, our system aims to recommend reliable and worthy information in real-time that can be used by editors to update the page.

The *reliability* of a web domain exists on a spectrum and indicates the degree to which the information presented on a website is accurate, trustworthy, and free from errors or bias. The reliability of web sources is a diverse topic. Sources and methodologies that are reliable for one platform may not suit other platforms. For e.g., google page rank can't be directly assumed as a good ranking scheme for Wikipedia. So, we design a reliability scheme of web domains with Wikipedia as focus.

We define the *worthiness* of a sentence as how fit it is to be added to the Wikipedia page being processed. Sometimes reliable information might not be worthy. Consider for example, "Today, Messi and Ronaldo played against each other." This sentence might be reliable and sensational but not worthy for adding to Wikipedia's informative pages.

3 Methods

Our system consists of three components: a source reliability scheme, fetching the latest information from reliable sources, and finally, verifying their worthiness.

3.1 Where to Get: Reliable Sources

To design the source reliability scheme, we compute the following four features from the edit logs and references of Wiki pages. These features are extracted corresponding to each reference from each page in a given Wikipedia category of interest.

$F1$. **Editor Count:** The Editor count indicates how many editors believe that information from a particular reference is reliable and worthy.

$F2$. **Coverage Relevancy:** This score determines the coverage of information on the reference page with respect to the information on the corresponding wiki page. We used the cosine similarity of corresponding Bert embeddings.

$F3$. **URL Domain Count:** The URL domain count (Pochampally and Karlapalem, 2017) a total number

of times a domain is referred in all pages that belong to the Wikipedia category of interest.

$F4$. **Diversity Score:** This score counts the number of unique wiki pages that contains a reference entry corresponding to a URL domain.

These features capture different aspects of the relevance of information sources, and complement each other in defining the overall reliability of URL domains.

3.2 Feature Importance in Reliability Scheme

As we lack the ground truth, we analyze well-known unsupervised feature importance algorithms to assign each feature a specific weight. These algorithms include Laplacian feature importance algorithms (He et al., 2005), Spectral feature importance algorithms (Zhao and Liu, 2007), and PCA (Principle Component Analysis) (WOL, 1987).

These unsupervised algorithms capture different complementary aspects, including locality-preserving power, local and global structure, and variance. Yet we found all these algorithms to agree on the same relative feature importance, as $F1 \geq F3 \geq F2 \geq F4$. This analysis helps to understand the data and assign feature weights to define our reliability scheme. However, the editor can manually pass these weights as a hyper-parameter based on domain knowledge also.

3.3 What to Get: Reliable Sentences

We use two ways to obtain reliable content. First, the editor selects some top k domains as reliable (using the reliability scheme above) and extracts content from these pages. Second, in order to retrieve the latest information, we query for the notable entity corresponding to the wiki page title using google search API. If the web domain is not at all present in our reliability scheme results, we discard it. Otherwise, we include it.

Next, we compute the reliability score for each sentence as the sum of the reliability scores of web domains whose pages contain that sentence. To determine the presence of a sentence S_i in a page P_j, we use a threshold α on the cosine similarity of sentence Bert representations.

$$Score_j(S_i) = \begin{cases} R_l, & \text{if cosine similarity}(S_i, S_{jm}) \geq \alpha \\ 0, & \text{otherwise} \end{cases}$$

This equation assigns the reliability score R_l of a web domain l to a sentence S_i if it is present in a page P_j of the web domain. S_{jm} represents the m^{th}sentence on page P_j, having maximum similarity with S_i.

The cumulative reliability score for the sentence S_i is then the sum of the reliability scores across web domains whose pages contain that sentence:

$$Score(S_i) = \sum_{j=1}^{j=t} Score_j(S_i)$$

3.4 What to Get: Worthy Sentences

After passing a threshold for reliability, sentences must pass a check for worthiness, as described in Section 2. We train a classifier to determine if a sentence is worthy enough to include in Wikipedia. The positive samples for training are all reference sentences whose cosine similarity with corresponding Wikipedia page sentences is greater than a threshold, while all other sentences are negative. This is done because if a particular information was not added by the editor from a known web URL to the wiki page, then it is unlikely to be worthy.

4 Results

The dataset for designing the reliability scheme was formed using the four features described in Section 3.1, extracted for the Wikipedia category of Indian Hindi Actors. Feature weights were set based on the method in Section 3.2. The top reliable domains are seen in Table 1 The classifier that determines the sentence addition score has been tested. The results in Table 2 show the sensitivity (recall) score for different cosine similarity thresholds.

5 Discussion/Conclusions

As our model aims to recommend reliable and worthy information to the editor for wiki updates, our recall should be high as we would not want to miss any worthy information. We experimented classifier with a threshold(β) above 0.4, but discarded it as it produced unbalanced data. As seen in the results, our model's recall scores are very high (98% for $\beta = 0.4$), showing that it is able to extract and recommend worthy sentences.

References

[He et al.2005] Xiaofei He, Deng Cai, and Partha Niyogi. 2005. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems*.

[Matheson2020] Rob Matheson. 2020. Automated system can rewrite outdated sentences in wikipedia articles.

[Pochampally and Karlapalem2017] Yashaswi Pochampally and Kamalakar Karlapalem. 2017. Notability determination for wikipedia. WWW '17 Companion.

[WOL1987] 1987. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*.

[Zhao and Liu2007] Zheng Zhao and Huan Liu. 2007. Spectral feature selection for supervised and unsupervised learning. Association for Computing Machinery.

Domain Name	URL Domain Count	Editor Count	Coverage Relevancy Score	Diversity Score	Reliability Score
Indiatimes	1895	83258.63	159.55	393	17.66
Indianexpress	756	53102.38	134.44	261	10.31
Hindustantimes	750	58042.94	109.23	239	10.11
..	..	..	..	..	..
..	..	..	..	..	..
Manikarnikaiff	1	2.0	0.0	1	0.000074

Table 1: Reliability Scheme and Score

Cosine Similarity Threshold (β)	Recall (Sensitivity)
0.2	94.75
0.4	97.6

Table 2: Sensitivity measure with different thresholds

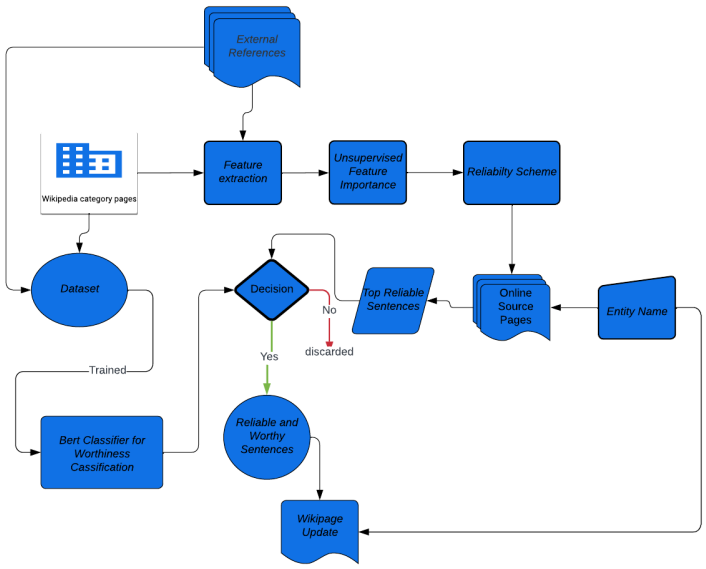

Figure 1: System Architecture