ISSN 1870-4069

WikiAug: Augmenting Wikipedia Stubs by Suggesting Credible Hyperlinks

Ayesha Siddiqa, Ashish V. Tendulkar,
Sutanu Chakraborti

Indian Institute of Technology Madras,
India

{m.ayeshasiddiqa, ashishvt}@gmail.com
sutanuc@cse.iitm.ac.in

Abstract. Wikipedia is the ubiquitous resource for knowledge acquisition for humans and widely used systems like Apple's Siri, IBM's Watson and Google's Knowledge Graph. Thus, it is crucial that Wikipedia's articles are timely and accurate. Currently, Wikipedia articles are created and maintained by volunteers who may not be experts of the field. Consequently, many related concepts and relevant literature pointers may be missed by the editors. Furthermore, owing to the huge size and growth rate of Wikipedia, a manual search for information and maintenance process is unworkable in the long run. Our proposed approach *WikiAug* attempts to fill in these gaps and recommend concepts which are missing from Wikipedia articles (more specifically Stubs). To recommend concepts to Wikipedia articles, we rely on external knowledge retrieved by a search engine and semantic information like category labels and structure available in Wikipedia articles. This semantic knowledge adds new dimensions to the state-of-the-art approaches.

Keywords: Hyperlinks, link augmentation, wikipedia augmentation, explicit semantic analysis.

1 Introduction

Wikipedia has emerged as a reliable, comprehensive and authoritative encyclopedia on the Web. As of today, the English Wikipedia contains *5* million articles which contain rich semantic information in the form of text content, hyperlinks etc. Typically these links are links to other Wikipedia articles and external websites which contain related information. Carefully placed hyperlinks present users with new related concepts[1] and are essential for content discoverability. Widely used applications which rely on Wikipedia for their knowledge acquisition are Google's Knowledge Graph and Apple's Siri system. Knowledge acquired from Wikipedia is harvested and utilized in building knowledge bases like YAGO [19], DBpedia [4], and has interesting applications in Text categorization [20], Named entity disambiguation [8], and Entity ranking [9]. Therefore, it is essential that Wikipedia's content is up to date and accurate.

[1] We will be using the terms *references*, *concepts* and *links* interchangeably

Wikipedia volunteers classify articles into seven categories: *FA*, *GA*, *A*, *B*, *C*, *Start* and *Stub*[2] respectively. According to statistics, 88% of the Wikipedia articles belong to the stub and start category. However, given the huge size and growth rate of Wikipedia, a manual revision and maintenance process is unworkable in the long run. The problem is not only limited to Wikipedia but also a critical issue for collaboratively built resources like Open Directory Project (ODP)[3]. Therefore, it is crucial to develop automatic link and content suggestion methods which aid editors discover related content and maintain the rich link structure of Wikipedia.

We enrich Wikipedia articles by recommending hyperlinks. More specifically, we address the following research problem: Given the introductory content, title, and semantics of a Wikipedia article like category and structure, can we suggest concepts which could help editors in discovering content related to them? We recommend links (concepts) and leave the choice of adding content from these links to editors. We envisage that our concept suggestions can be motivation for the new editors to write the articles without necessarily having enough domain expertise in the related topics they are writing or editing.

Our proposed method is motivated by how humans search for information. We search for information related to the stub by formulating queries against a search engine. Our central assumption here is information related to stub article is available on the web and can be retrieved by the search engine. The retrieved web content is used to provide concept suggestions for the stub.

Apart from the other methods which suggest missing links, we also, formalize the three properties of an appropriate reference, namely, suggested references should: (1) be relevant to the stub, (2) cover representative (diverse) subtopics of the stub, and (3) are obtained from authoritative sources on the web. Our proposed method attempts to ensure these properties while providing recommendations. For this work, we have limited our suggestions only to other Wikipedia articles. In summary, our essential contributions are: (1) We propose a novel method for augmenting the hyperlink structure of Wikipedia articles. (2) We propose an automated evaluation method which leverages Wikipedia's revision history.

2 Background

Explicit Semantic Analysis (ESA)

In order to obtain a meaningful interpretation of text, we use Explicit Semantic Analysis (ESA) proposed by (Gabrilovich and Markovitch 2007) [6]. ESA uses Wikipedia as its source of world knowledge and was proposed to estimate semantic relatedness between two text fragments. ESA takes a text fragment as input and returns a list of Wikipedia concepts as output which are weighted by the relevance of the concept to the text. The central hypothesis based on which ESA works is: *each article in the Wikipedia corresponds to a single concept.* An inverted index containing a mapping from words to Wikipedia articles that contain them is pre-built and stored as a preprocessing step.

[2] en.wikipedia.org/wiki/Wikipedia:Stub

[3] en.wikipedia.org/wiki/DMOZ

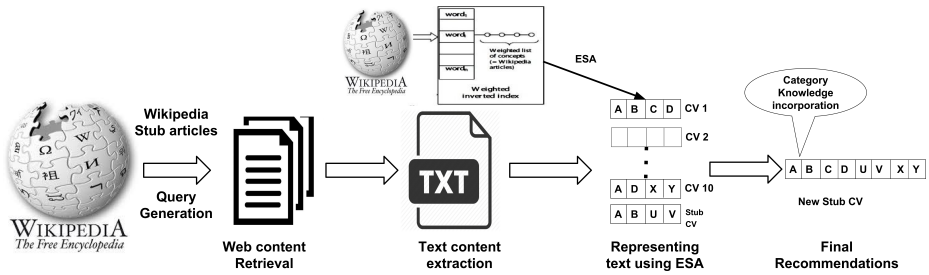

Fig. 1. WikiAug: System architecture.

To obtain the ESA representation of a text fragment, for each word in the text, ESA inverted index is searched and then the corresponding concepts containing that particular word are retrieved. These concepts are combined to form the weighted concept vector, where weights correspond to the TFIDF value of the word in the ESA article corresponding to the concept. The list of concepts are ordered by the weight to get the final ESA representation.

3 Problem Formulation

In this paper, we study the problem of suggesting Wikipedia articles (concepts) that contain relevant information to add to Wikipedia stubs. We formalize the task as follows. Given a Wikipedia stub S and a set of all Wikipedia articles W, we have to decide for each $w_c \in W$ whether it is relevant for improving the content of Stub S. More formally, we have to estimate:

$$P(rel|w_c, S), \tag{1}$$

where rel is the relevance of document w_c with respect to stub S. An article is considered relevant and a valid suggestion if it can enhance the current content of stub S. Note that we are limiting our suggestions to Wikipedia pages but they can be any other web pages providing the description of the stub.

4 WikiAug (WA)

Our proposed approach for link augmentation is called *WikiAug (WA).* Figure 1 shows the system architecture of *WikiAug.* Our approach is motivated by how humans search for information. We observe that it is typical of humans to search for information using the stub title as a *search keyword* or as a *query* against the search engine.

Our central assumption for this work is, information related to the stub is available on the Web and can be retrieved by the search engine. $WikiAug$ system takes a Wikipedia stub title S as input and returns a list of missing Wikipedia concepts L. We achieve this by comparing S against authoritative sources on the Web.

For a given Wikipedia article S, our system deals with two key issues: (1) How do we find authoritative sources for S from the Web? and (2) How do we find missing concepts in S?

As shown in Figure 1, we propose a five-step procedure *WikiAug (WA)* for link augmentation of stubs. These steps are described in the Table 1. Concepts suggested by our method are classified as relevant or irrelevant based on the ground truth links. If a link suggested by our method is present in the ground truth links set it is considered as a *relevant* one else an *irrelevant* one. Our link suggestions for a query are given in the Table 3.

4.1 Definitions

Wikipedia Article Graph (WAG). Is the graph constructed with nodes as articles and edges representing hyperlinks between the Wikipedia articles.

Wikipedia Category Graph (WCG). Wikipedia categories[4] are organized in a taxonomy like structure called the *Wikipedia Category Graph (WCG)* which captures hyponymy or meronymy relations. Wikipedia category may contain subcategories which further contain Wikipedia articles. However, since Wikipedia does not enforce strict guidelines of taxonomy; the graph may contain cycles or disconnected subcategories. However, these cycles or disconnected subcategories do not influence our algorithm's performance since we do not use any graph theory algorithms in our approach.

Wikipedia Subtopics (Aspects) or Structure. We define structure[5] as the representative subtopics information of a stub article (topic). The central idea here is that any Wikipedia article contains similar sections in similar Wikipedia articles. This motivates us to leverage this structure to cover representative subtopics of the stub article (Details in subsection 4.4). For instance, a query like *Donald Knuth* contains similar sections like *Education, Academic life, Research, Awards and achievements* and *Books and Publications* which are similar to sections in other similar articles.

4.2 Content to Capture Context of the Query (Co)

To mitigate the problem of irrelevant recommendations we consider context of the stub article (query). This is the same idea as suggested in ESA (Gabrivolich et al. 2007) [6]. They hypothesize that ESA may perform inherent disambiguation when provided with the sufficient context of the stub query. To obtain context, we represent the stub article text in ESA Concept Vector and obtain top 10 concepts. These concepts are appended to the stub title and this forms the set of reformulated queries given in Table 2 for Web content retrieval (Refer to the Table 2 for examples of reformulated queries).

4.3 Classification Using Category Knowledge (Cat)

Furthermore, analysis of suggestions by WA and WA+Co gave us insights that we need domain knowledge to provide meaningful recommendations. This domain knowledge is available in the form of category labels.

[4] en.wikipedia.org/wiki/Category

[5] en.wikipedia.org/wiki/Help:Section

Table 1. WikiAug method (WA).

1. **Query Generation:** To search for information on the web, we formulate queries using the stub title. For example, for suggesting links to the stub article on *Michael Jordan*. The query formulated using the stub title is *Michael I. Jordan*. Modified queries are formulated to improve efficacy of WA by the below-mentioned approaches. The following modifications are done in this step for WA variants:
   - Title (WA)+Co,
   - Title (WA)+Co+S,
   - Details of. Structure extraction and Category knowledge extraction are explained in Subsection 4.4 and 4.3 respectively.
2. **Web Content Retrieval:** The query generated from the above step is used to fetch top 10 URLs (search results) from a search engine. We use Google for obtaining the results. These results form the set of relevant Web content (external Web articles) used for suggestion,
3. **Text Extraction:** Web content obtained from the previous step is typically available as web pages. Web pages, in general, contain text snippets in between the HTML tags along with the noise. Therefore, it requires cleaning to retain only the relevant information. Removal of irrelevant content is done using Boilerplate detection[10]. This technique classifies the text snippets as relevant if they contain information related to the body of the article and irrelevant if it is website related content (e.g., header and footer or advertisements content present in a webpage),
4. **Representing Text in ESA Concept Vector:** The resultant text from the above step is given as input to ESA. As mentioned earlier, we obtain *Concept vector* or *Meaning vector* using ESA. Each component of this $Concept\ vector$ is the title of a Wikipedia article. We represent both stub text and the text extracted from external web articles in ESA concept vectors,
5. **Final Recommendations:** From the above step, we obtain concept vectors for the stub and the other external Web articles. Now, we compare Wikipedia article concept vector against the web article concept vectors by looking for the concepts covered. The concepts which are absent in the stub but covered in other articles are given as recommendations to the stub.
   - The following step is for variants of WA named as WA+Co+Cat, WA+Cat+S and WA+Co+Cat+S.:
   - In this step, we classify the WA suggestions with the help of Wikipedia category knowledge by posing this as a classification problem. Further details in Subsection 4.3.

We use these category labels as our class labels and our suggestions as the test instances to be classified by the classification task. We approach this classification task by posing it as a *Multi-Class classification* problem. Thus, we hypothesize that: Semantic features like category label available in Wikipedia articles are useful for providing meaningful suggestions. Refer to the Table 2 for examples of reformulated queries. We prefer multi-class classifiers over binary classifiers because they give us a more balanced distribution when compared to merging all non-relevant recommendations into a single category. To perform this classification, we leverage existing work on text classification by (Chang et al. 2008) [5]. This classification is performed without any labelled training data, utilizing the world knowledge available in ESA, and by understanding of the labels (i.e., mapping documents and labels to a semantic space).

Algorithm 1: Structure or aspect extraction.

input : Wikipedia stub category: c, Wikipedia articles: W, parameter: k
output: Wikipedia structure (or aspects): A
$C \leftarrow retrieveArticles(W, c)$
$H_C \leftarrow extractSectionHeadings(C)$
$aggregateSectionHeadings(H_C)$
for *each* $heading(h) \in H_C$ **do**
  $S_C \leftarrow retrieveSections(H_C, h)$
  $A \leftarrow FrequentSections(S_C, k)$
end

This classification is performed in two steps:

1. In the first step, we represent both labels and documents in a semantic space that allows one to compute the semantic similarity between a document (stub) and a potential label.
2. In the second step, we use the similarity computed in the first step to learn a machine learning classifier for classifying recommendations.

4.4 Wikipedia Structure (Aspects) Extraction(S)

As mentioned in section 4.1, Structure (S) contains key pieces of information available in a Wikipedia article. We extract structure with the help of section titles of Wikipedia articles. We accumulate these section titles from similar articles belonging to the Stub category. Our algorithm for structure (aspect) extraction is given in Algorithm 1. We adapted our structure extraction algorithm from the prior work by (Reinanda et al. 2016) [16], (Sauper et al. 2009) [17] and (Banerjee et al. 2015) [2].

As shown in the algorithm, we first, retrieve Wikipedia articles of all the stub queries belonging to their corresponding category c. It can be observed that it is typical of Wikipedia articles that they belong to several categories. All possible categories labels for any Wikipedia page are listed at the bottom of the page. Among the possible categories, we choose the most relevant stub category as input to our Structure extraction algorithm. To select the most relevant stub category, we estimate the ESA semantic similarity between the possible category labels and stub title. The category with the maximum similarity value is chosen as the most relevant one.

Further, we extract the section headings within the articles belonging to the selected most relevant category. For each section heading, we remove the stop words and aggregate the section heading into one section heading, this is our aggregate section headings step. We then take the k most frequent sections as Structure (or Aspects) from the extracted section headings. Refer to Table 2 for reformulated queries.

5 Empirical Evaluation

5.1 Datasets Description

We performed experiments on datasets belonging to the Wikipedia categories *Machine Learning, Physics, Computing, and Environment*.

Table 2. Reformulated queries for the query *Michael I. Jordan*.

Method	Reformulated queries
WA	Michael I. Jordan
WA+Co	Michael I. Jordan and Association for the Advancement of Artificial Intelligence, Michael I. Jordan and Computer scientist, Michael I. Jordan and Daniel Walker Howe, Michael I. Jordan and University of California Berkeley, and Michael I. Jordan and Nonparametric regression
WA+Co+Cat	same as WA+Co
WA+S+Cat	Michael I. Jordan and Education, Michael I. Jordan and Career, Michael I. Jordan and Academic life, Michael I. Jordan and Biography, Michael I. Jordan and Research, Michael I. Jordan and Honors and awards, Michael I. Jordan and Notable Students, Michael I. Jordan and Roles, Michael I. Jordan and Positions of responsibility and Michael I. Jordan and Books and Publications
WA+Co+Cat+S	Combined set of queries from WA+Co and WA+S+Cat

We curated these datasets by extracting articles using the Wikipedia API[6]. Given a Wikipedia category, this API recursively extract articles present in the category hierarchy that can be reached by the crawler using top-down traversal and allows us to download articles in XML format.

We then extract text content and hyperlinks available in these articles using a python script called *WikiExtractor*[7]. Our curated datasets are Wikipedia articles belonging to the years 2008 and 2015 from the above mentioned Wikipedia categories. From each category, we consider only the Wikipedia articles which got promoted to higher quality grades (relatively Non-Stub) in 2015 from Stubs in 2008 as our queries for link augmentation.

5.2 Classification Using Category Knowledge: Experimental Setup

As already mentioned, our classification assumes no training data is available to us, the only information available is the category of the labels. The efficacy of our approach lies in the rich semantic representation like ESA representation. We use the ESA implementation provided in Descartes library.[8]

To evaluate the effectiveness of the ESA representation, we use for each text fragment concept vector representations of size 10, 50 and 100 concepts. We then use K-Nearest neighbour classifier to perform classification and $K = 20$ set empirically for all the datasets mentioned above. All the results presented here are the average of ten trials. Accuracy of the classifier on the test data is *65.7%* using *BOW-NN* and *85.3%* using *ESA-NN*.

Even though we have used ESA for text representation recently there have been approaches for learning Word embeddings like *Word2vec* proposed by (Mikolov et al. 2014) [13] and have been widely accepted in the NLP community. These approaches are for representing words in terms of vectors and the embeddings are typically trained by neural networks. A thorough comparison of the suitability of several different word embeddings for classification without training labels called as *Dataless Classification*

[6] en.wikipedia.org/wiki/Special:Export

[7] medialab.di.unipi.it/wiki/Wikipedia_Extractor

[8] cogcomp.cs.uiuc.edu/software/descartes/descartes-0.2/doc/README.html

Table 3. Recommendations given by WikiAug and its variant methods for the query *Michael I. Jordan*.

* Only a few irrelevant recommendations are shown for the sake of brevity.	
Method	**Recommendations**
WA	**RELEVANT:** Non-parametric regression, Semi parametric regression, Layers, non-parametric, nonlinear, pehong, chen, distinguished, david, heckerman, kearns, marina, meila, Nonlinear system, Non-parametric statistics, Layering, Generalization error
	IRRELEVANT: Susan L. Graham, Chicago Bulls season, Sam Michael, Jordan, James R. Jordan, Sr., Sean Chen (politician), Steve Chen, The Jordan Rules (book), Sean Chen (artist), Chicago Bulls season, Jordan–European Union relations, Rugby union in Jordan **
WA+Co	**RELEVANT:** Association for the Advancement of Artificial Intelligence, Semi-parametric regression, Non-parametric regression, Non-parametric statistics, Regression analysis, Artificial Intelligence: A Modern Approach, Polynomial regression, Statistical model, Non-parametric Bayesian methods, Markov Chain Monte Carlo, non-parametric analysis, probabilistic graphical models
	IRRELEVANT: Daniel Walker Howe, Shauna Howe, Timothy O. Howe, Lord Howe Island Airport, David J. Howe, Howse Pass, Jeremy Howe, Jordan, Jordan-European Union relations, Stephen Woolley, Rational agent, Sam Michael, Jordan, James R. Jordan, Sr., Piero Scaruffi, 1990-91 Chicago Bulls season, Sam Michael, Jordan
WA+Co+Cat	**RELEVANT:** Association for the Advancement of Artificial Intelligence, Semi-parametric regression, Non-parametric statistics, Non-parametric regression, Non-parametric statistics, Artificial Intelligence: A Modern Approach, Polynomial regression, Statistical model, Non-parametric Bayesian methods, Non-parametric analysis, Markov Chain Monte Carlo, Probabilistic graphical models, Non-parametric analysis
	IRRELEVANT: Geoffrey Hinton, Peter Norvig, James R Jordan
WA+Cat+S	**RELEVANT:** Louisiana State University, University of California San Diego, David Rumelhart, Professor, University of California Berkeley, University of California, Berkeley College of Letters and Science, Recurrent neural networks, Bayesian parametric networks, Non-parametric statistics, Non-parametric regression, Non-parametric statistics, Polynomial regression, Statistical model, Non-parametric Bayesian methods, Non-parametric analysis, Markov Chain Monte Carlo, Probabilistic graphical models, Dirichlet Process, ACM-AAAI Allen Newell Award, Association for Computing Machinery, Association for the Advancement of Artificial Intelligence, Artificial Intelligence: A Modern Approach, Andrew Ng, Zoubin Ghahramani, Francis Bach David Blei, Eric Xing, Martin Wainwright, Ben Taskar and Yoshua Bengio
	IRRELEVANT: Kent A Jordan, Peter Norvig, Piero Scaruffi, MIT Center for Collective Intelligence, Jordan, Tom M Mitchell, Miller Institute, Igor Aleksander, Artificial brain, Sigma Xi, Bureau of Intelligence and Research, Edward Feigenbaum, Artificial Intelligence: A Modern Approach, Mike Jackson (systems scientist), Minds and Machines, Sam Michael, John Canny, Geoffrey Hinton, James R Jordan, Sr, Commonsense knowledge base, Seed AI
WA+Co+Cat +S	**RELEVANT:** Louisiana State University, University of California San Diego, David Rumelhart, Professor, University of California Berkeley, University of California, Berkeley College of Letters and Science, Recurrent neural networks, Bayesian parametric networks, Non-parametric statistics, Nonparametric regression, Non-parametric statistics, Artificial Intelligence: A Modern Approach, Polynomial regression, Statistical model, Nonparametric Bayesian methods, Nonparametric analysis, Markov Chain Monte Carlo, Probabilistic graphical models, Dirichlet Process, ACM AAAI Allen Newell Award, Association for Computing Machinery, Association for the Advancement of Artificial Intelligence, Andrew Ng, Zoubin Ghahramani, David Blei, Eric Xing, Martin Wainwright, Yee Whye Teh, Ben Taskar and Yoshua Bengio
	IRRELEVANT: Kent A Jordan, Artificial Intelligence: A Modern Approach, Peter Norvig, Geoffrey Hinton, Piero Scaruffi, Tom M Mitchell, Miller Institute, Igor Aleksander, Artificial brain, Sigma Xi, Bureau of Intelligence and Research, Common sense knowledge base

was given by (Song et al. 2014) [18]. Their comparison study validates that out of all the semantic similarity measures considered for *Dataless classification*, ESA performs better.

Since our ultimate goal was concept recommendations for Wikipedia articles utilising ESA representation seemed straight forward approach as the output of ESA algorithm is set of Wikipedia concepts which can be directly utilized as concept recommendations.

5.3 Link Recommendation Evaluation

We demonstrate the effectiveness of our *WikiAug* method and its variants by evaluating it on Wikipedia articles.

Groundtruth. Wikipedia allows access to complete revision history of its articles. We accessed from the revision history, links which got added to the stub articles over the years 2008 to 2015 and these links form our set of ground truth links. The set of ground truth links is incomplete because many relevant and important links might have been missed but may allows us to gain useful insights.

Evaluation Measures. To evaluate our recommendations, we defined three measures, i.e., Precision (P), Recall (R) and F_1-score. For our evaluation, a true positive recommendation is a link suggested and added to the corresponding 2015 Wikipedia article. In a similar fashion, a false negative and false positive have been defined. Average Precision and Recall for each Wikipedia article are calculated and averaged over the number articles considered for evaluation and the corresponding F_1-score is also calculated accordingly.

Baseline. We propose our own baseline method which takes help of Wikipedia category tree for providing recommendations. As already described in the Wikipedia category graph (WCG), for a category C, there can be subcategories $S_1, S_2 \dots S_n$ and within each subcategory there can be many Wikipedia articles.

Suppose $W_1, W_2 \dots W_n$ belong to subcategory S_1 and W_1 is a Stub article. Our baseline method recommends all Wikipedia articles at the same level (siblings) as W_1 to the stub W_1. That is essentially our baseline method is doing a Depth first traversal on Wikipedia category tree to reach the stub article node and then a Breadth first traversal from the stub node to obtain all the nodes present at the same level as stub. This way we end up obtaining the sibling nodes of stub and these form the set of our baseline suggestions.

The Wikipedia category tree has the property of all similar articles being grouped under the same category or subcategory by the Wikipedia editors (humans). Therefore, articles at the same level will be highly relevant to each other and this leads to strong recall of our baseline method. Although our baseline has a strong recall, it suffers from poor precision, since the diverse information scattered across different subcategories, categories and at different depths is not captured while providing the suggestions.

We compared precision, recall obtained by our baseline method with the methods proposed above. These results are given in Table 4. As observed in Table 4, incorporating category knowledge leads to higher precision. This trend is observed because we do not suggest the links which are irrelevant or farthest in meaning to stub.

Table 4. Precision (P), Recall (R) and F-score (F_1) values obtained from Baseline and WA and it's variants. Highest F-score (F1) values obtained are in Bold.

	Computing			**Machine Learning**			**Physics**			**Environment**		
Method	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Baseline	0.02	0.28	0.03	0.02	0.26	0.04	0.04	0.39	0.07	0.04	0.48	0.07
WA	0.19	0.08	0.11	0.07	0.06	0.06	0.17	0.11	0.13	0.29	0.14	0.19
WA+Co	0.21	0.11	0.15	0.14	0.07	0.10	0.23	0.16	0.19	0.33	0.15	0.20
WA+Co+Cat	0.71	0.16	0.26	0.69	0.15	0.25	0.71	0.17	0.27	0.79	0.21	0.33
WA+Cat+S	0.62	0.46	0.53	0.58	0.34	0.43	0.69	0.59	0.64	0.70	0.62	0.66
WA+Co+Cat+S	0.73	0.48	**0.58**	0.63	0.37	**0.47**	0.72	0.61	**0.67**	0.75	0.64	**0.69**

Table 5. Precision (P), Recall (R) and F-score (F_1) values obtained from (West at al. 2009 - 10)'s approach and WikiAug (WA+Co+Cat+S) approach.

	West et al. (2009)			**WikiAug**		
Category	P	R	F_1	P	R	F_1
Computing	0.15	0.20	0.17	0.73	0.48	0.58
Machine Learning	0.11	0.19	0.14	0.63	0.37	0.47
Physics	0.17	0.26	0.21	0.72	0.61	0.67
Environment	0.23	0.32	0.27	0.75	0.64	0.69

However, this doesn't contribute to increase in recall. To improve recall of our method, we incorporated structure knowledge. This lead to retrieval of diverse links which are effective in covering various aspects of the stub. We further present our quantitative results by comparing our method's precision, recall and F_1-score with the values obtained by (West et al. 2010) in the Table 5.

6 Related Work

The link prediction problem in information networks like Wikipedia is posed in many variants. A comprehensive study of this problem was performed by (Nowell et al. 2007) [11].

Since augmentation of Wikipedia stubs is the broad goal of our work, we have used Wikipedia as our primary dataset for evaluation. In this section, we briefly describe other methods which proposed solutions for link prediction and augmentation in Wikipedia.

6.1 Link Prediction in Wikipedia

In this section, we compare and contrast our work with other approaches for predicting missing hyperlinks from Wikipedia. Automated methods to identify missing hyperlinks from Wikipedia were extensively studied in the past literature [12, 14, 7].

Table 6. Top suggestions given by WikiAug for the query eurozone crisis.

Human added	**West et al. 2009**	**WikiAug**
1. European debt crisis	1. European central bank	1. European debt crisis
2. European Union	2. Inflation	2. Causes of the European debt crisis
3. European Central Bank	3. OECD	3. European Fiscal Compact
4. International Monetary Fund	4. Eurobarometer	4. European Union
5. Fiscal Union	5. Single market	5. Greek government debt crisis
6. European Stability Mechanism	6. Gross domestic product	6. European Stability Mechanism
7. Outright Monetary Transactions	7. European commission	7. Policy reactions to the Eurozone crisis
8. European Financial Stability Facility	8. Motion of no confidence bank	8. Economic and Monetary union of the European union

We classify these methods broadly into two categories based on the type of solutions they provided. The first group of methods proposed by (Mihalcea et al. 2007) [12], (Milne et al. 2008) [14], and (Meij et al. 2012) [7] solved the problem of predicting the missing out-going links from a given Wikipedia article. These methods process arbitrary text documents and predict outgoing links. They leverage the textual content and graph structure of Wikipedia for predicting missing links.

The second group of methods (West et al. 2009) [21], (Adafre et al. 2007) [1], and (Noraset et al. 2014) [15] solved the problem of predicting future outgoing links to Wikipedia articles. These methods take a Wikipedia article as input and utilize already existing links or perhaps text content to predict future outgoing links.

6.2 Content Augmentation

Content Augmentation deals with adding textual content to incomplete or simply stub articles. In this subsection, we briefly describe past literature on content enrichment. (Sauper et al. 2009) [17] proposed solution for the problem of populating Wikipedia pages obtaining content from Web search engines. They extract web content formulating queries with the help of article title and structure extracted from similar articles in a domain.

Recent work by (Banerjee et al. 2015) [2] and (Banerjee et al. 2016) [3] developed systems WikiKreator and WikiWrite which focused on populating content summaries for stub and red-link articles respectively. WikiKreator system proposed by (Banerjee et al. 2015) leverages category knowledge to suggest minimally redundant and more interpretable content summaries. In addition to this, WikiWrite system incorporates coherence.

6.3 Comparison to Previous Methods

To highlight the contributions of this work, we will now compare it with the existing approaches introduced in Section 6. (West et al. 2009) proposed an approach which outperformed the then the state of the art algorithm [14].

Table 7. Left: The 18 novel links (i.e., topics) human editors added to the Wikipedia article about *Computer Programming* between March 2009 and April 2010. Middle: The 18 top suggestions of West algorithm Right: The 18 top suggestions of WikiAug.

Human-added suggestions	West et al., suggestions	WikiAug suggestions
1. Troubleshooting	1. turing completeness	1. Outline of computer prog.
2. U.S. Air force	2. computer science	2. Pascal (prog. lang.)
3. Adacore	3. High - level prog. lang.	3. Callback (computer prog.)
4. Principle of linguistic relativity	4. Java (Prog. lang.)	4. APL (prog. lang.)
5. Card stock	5. Control flow	5. Automatic prog.
6. High - level prog. lang.	6. Haskell (prog. lang.)	6. History of prog. lang.
7. Temporary file	7. UNIX	7.Prog. tool
8. Memory leak	8. Lisp (prog. lang.)	8. Prog. lang.
9. Race condition	9. Ruby (prog. lang.)	9. Java (prog. lang.)
10. Ergonomics	10. Type system	10. Software
11. Maintainability	11. Subroutine	11. Functional prog.
12. Software bug	12. Comparison of prog. lang.	12. Julia (prog. lang.)
13. Vulnerability (Computing)	13. Difference engine	13. Application prog. interface
14. Scripting lang.	14. Python (prog. lang.)	14. Clarion (prog. lang.)
15. Measuring prog. lang. popularity	15. Z3 (computer)	15. Game prog.
16. 1947	16. Halting problem	16. Parameter (computer prog.)
17. The art of computer prog.	17. Abacus	17. Scheme (prog. lang.)
18. Gerhald Weinberg	18. UNIX - Like	18. Extreme prog.

(Adafre et al. 2007)'s method suggest relevant links based on the inlink similarity. Whereas, (West et al. 2009)'s method suggests relevant links based on the outgoing link similarity. Similar to their method we also identify relevant outgoing links. The above two methods capture structural similarity, i.e., the similarity based on the structure of the graph. These methods work with the underlying assumption that *Similar articles contain similar outgoing links*.

They identify similar articles with the number of shared outlinks as a similarity measure. But, this doesn't work effectively for stub articles since do not have the nice properties like strong connectivity to the other Wikipedia articles or sufficient text content. Thus, we devised an orthogonal similarity measure like text similarity (ESA similarity), Domain (Category) information and Structure information.

Previous methods discussed in related work are concept detectors, where as an approach proposed by (West et al. 2010) [22] is a concept (topic) suggester. Similar to theirs, our method is also a concept suggester. Consequently, our method suggestions are not limited to terms or phrases appearing in the current input text but for every possible candidate Wikipedia article. Other than Wikipedia's textual and hyper-textual content, we are exploiting rich semantics utilizing others dimensions like category knowledge and structure available in Wikipedia which previous methods have ignored. We hypothesize:

1. *Category (domain) knowledge is effective to provide suggestion of a concept that is closest in meaning to the stub article,*
2. *Structure of an article is helpful in covering various subtopics to ensure diversity in the article content*.

Efficacy of our approach lies in exploiting these dimensions for providing concept suggestions. While (West et al. 2010) restricted themselves with only outgoing link similarities, we observed that stubs articles contain few outgoing links compared to enriched articles. and it becomes challenging to identify similar articles using only outgoing link similarity. Thus, our proposed dimensions can complement their similarity measure for discovering concepts (or links). We present our qualitative results on a query in Table 6 and 7.

7 Conclusions and Future Work

In this paper, we present a novel approach to enrich Wikipedia articles by suggesting missing links. We provide recommendations using the semantics of Wikipedia articles like category knowledge, content and structure template. We outline implications of the approach beyond link suggestion: It can detect concepts (links) which a Wikipedia article fails to cover.

We believe that this work will inspire the application of similar techniques. As a part of future work, we note that Wikipedia contains different types of links like *External Links*, *References* and *See also*. We intend to investigate how can we modify our approach for providing link suggestions for these type of links.

References

1. Adafre, S. F., Rijke, M.: Discovering missing links in Wikipedia. In: Proceedings of the 3rd international workshop on Link discovery, pp. 90–97 (2005) doi: 10.1145/1134271.1134284
2. Banerjee, S., Mitra, P.: WikiKreator: Improving Wikipedia stubs automatically. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, pp. 867–877 (2015) doi: 10.3115/v1/P15-1084
3. Banerjee, S., Mitra, P.: WikiWrite: Generating Wikipedia articles automatically. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pp. 2740–2746 (2016)
4. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the web of data. Journal of Web Semantics, vol. 7, no. 3, pp. 154–165 (2009) doi: 10.1016/j.websem.2009.07.002
5. Chang, M., Ratinov, L., Roth, D., Srikumar, V.: Importance of semantic representation: Dataless classification. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI'08, pp. 830–835 (2008)
6. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, pp. 1606–1611 (2007)

7. He, J., de-Rijke, M.: An exploration of learning to link with Wikipedia: Features, methods and training collection. In: 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, pp. 324–330 (2009) doi: 10.1007/978-3-642-14556-8_32
8. Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP'11, pp. 782–792 (2011)
9. Kaptein, R., Serdyukov, P., de-Vries, A. P., Kamps, J.: Entity ranking using Wikipedia as a pivot. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management, pp. 69–78 (2010) doi: 10.1145/1871437.1871451
10. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the Third International Conference on Web Search and Web Data Mining, pp. 441–450 (2010) doi: 10.1145/1718487.1718542
11. Liben-Nowell, D., Kleinberg, J. M.: The link prediction problem for social networks. In: Proceedings of the 12th International Conference on Information and Knowledge Management, pp. 556–559 (2003) doi: 10.1145/956863.956972
12. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, pp. 233–242 (2007) doi: 10.1145/1321440.1321475
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, vol. 2, pp. 3111–3119 (2013)
14. Milne, D. N., Witten, I. H.: Learning to link with wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM'08, pp. 509–518 (2008) doi: 10.1145/1458082.1458150
15. Noraset, T., Bhagavatula, C., Downey, D.: Adding high-precision links to wikipedia. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 651–656 (2014) doi: 10.3115/v1/D14-1072
16. Reinanda, R., Meij, E., de-Rijke, M.: Document filtering for long-tail entities. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM'16, pp. 771–780 (2016) doi: 10.1145/2983323.2983728
17. Sauper, C., Barzilay, R.: Automatically generating wikipedia articles: A structure-aware approach. In: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 208–216 (2009)
18. Song, Y., Roth, D.: On dataless hierarchical text classification. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, vol. 28, pp. 1579–1585 (2014) doi: 10.1609/aaai.v28i1.8938
19. Suchanek, F. M., Kasneci, G., Weikum, G.: Yago: A core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706 (2007) doi: 10.1145/1242572.1242667
20. Wang, P., Domeniconi, C.: Building semantic kernels for text classification using wikipedia. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 713–721 (2008) doi: 10.1145/1401890.1401976
21. West, R., Precup, D., Pineau, J.: Completing wikipedia's hyperlink structure through dimensionality reduction. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1097–1106 (2009) doi: 10.1145/1645953.1646093
22. West, R., Precup, D., Pineau, J.: Automatically suggesting topics for augmenting text documents. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management, pp. 929–938 (2010) doi: 10.1145/1871437.1871556