Notability Determination for Wikipedia

Yashaswi Pochampally
DSAC, IIIT-Hyderabad
India
p.yashaswi@research.iiit.ac.in

Kamalakar Karlapalem
DSAC, IIIT-Hyderabad
India
kamal@iiit.ac.in

ABSTRACT

Being the ever-growing online encyclopedia, Wikipedia requires a keen investigation about which articles are to be included for it to maintain its indispensability. To prevent unnecessary articles from being included, official guidelines of Wikipedia demand these named entities to meet “notability” standards for their article inclusion. In this paper, we evaluate named entities for their notability by using reliability and entity salience features. Evaluations of our system provide evidence for the viability of our solution as an alternative to the manual decisions made by the reviewers for inclusion of an article using the notability rules.

Keywords

Wikipedia; Notability; Reliability; Entity Salience

1. INTRODUCTION

At present, English Wikipedia has over 5 million human edited articles and an estimated 15,000 new articles created per month[1]. Considering such a massive growth rate of Wikipedia, manual addition of these articles would eventually become a labour intensive task. Thus, there arises a need for an automatic Wikipedia-ensemble, a system which dynamically adds new Wikipedia articles for named entities which meet the notability criteria. Notability is a test used by editors to decide whether a given named entity warrants its own article in Wikipedia.

For any new named entity, such an ensemble automatically checks whether a Wikipedia page already exists and if there is no such page, it creates a new one after checking whether the named entity meets the notability criteria. Hence, our automation ensemble would require automation of

1. *notability of the named entity*: given a named entity, the system should automatically decide whether the entity is notable or not.
2. *content to be included in the article*: the system should automatically extract information from web that has to included in a Wikipedia article for a given notable named entity.
3. *reliability of the content in the article*: system has to make sure that the auto-filled Wikipedia information for these named entities is extracted from reliable sources.

For automation of the reliable content of an article, there is ongoing research. A few such works include [6], [8] and [5]. As there has been work towards automating the creation of viable wikipedia articles, there arises a need to automate the notability of the named entity as well, for completing the automation ensemble. In this paper we target that notability automation.

Notability of a named entity depends on various factors like subject-specific guidelines, neutral point of view, verifiability, etc as described in its official Wikipedia page[2]. We focus on automating two major notability criteria among them which include availability of its reliable content in web and the coverage of the named entity in such content. For instance, let us consider existence of the entry in imdb.com and the coverage of the actor name in such an article as an indicators for the reliability and coverage for actors. We consider the following example actors.

1. Saif Ali Khan: has entry in imdb.com[3] with information about him covered in the content.
2. Rajwinder Deol: has no entry in imdb.com
3. Karikolraj: has his entry in imdb.com[4] but the information in the article does not have significant coverage of him.

Based on our indicators for notability, we can say that Saif Ali Khan is notable and the other two are not. The verification of our result would be done by checking their entry in Wikipedia, where Saif Ali Khan has a dedicated article[5] while the other two do not.

[1] https://en.wikipedia.org/wiki/Wikipedia:Modelling_Wikipedia's_growth

[2] https://en.wikipedia.org/wiki/Wikipedia:Notability

[3] http://www.imdb.com/name/nm0451307/

[4] http://www.imdb.com/name/nm8038256/

[5] https://en.wikipedia.org/wiki/Saif_Ali_Khan

WWW'17 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3053361

We have used the wikipedia category, Indian Film actors as an example for demonstrating the approach. It can also be extended to other categories having named entities, whose information is included in some common web-domains, which would be used as features in our approach. For example, articles in wikipedia category: Software Companies have information about its named entities in websites like linkedin.com, crunchbase.com, etc. Similarly, articles in wikipedia category:Films have information in websites like imdb.com. However, our system might not work efficiently for wikipedia categories like Concepts, theories, etc, where information cannot be extracted from definitive set of web-domains. This is a possible limitation to our approach.

The subsequent discussion of this paper is organized as follows. We discuss the related work in Section 2. In Section 3, we discuss the details about our approach and the features (indicators) used for automating the notability. Section 4 presents the experiments and results over various features. Finally, we conclude in Section 5.

2. RELATED WORK

[7] has proposed a system that provides personalized recommendations to editors for creation of articles that exist in one language but are missing in another. They ranked these missing articles using features like page counts and quality of the article in existing languages.

However, to the best of our knowledge there has been no research done in the direction of automation of wikipedia notability which would enhance the work done on automatically generating wikipeidia aritles like [6], [8] and [5] by becoming a pre-processing step for the automation ensemble discussed in section 1.

Using notability features like reliability and entity salience rather than leveraging cross-language wikipedia content sets our approach for suggesting articles that warrant wikipedia articles apart from [7]. Not using cross-language content ensures that we do not suggest some entity with no online information in english language as a notable entity to the ensemble engine because the approaches [6], [8] and [5] do not handle cross-language content for the new article generation. However investigating reliability and entity salience features over cross-language online data and extending the article generation approaches to support cross-language content using language translators would be a good future direction for improvement to our automation ensemble.

3. STRATEGY FOR NOTABILITY DETERMINATION

Given an entity name belonging to a Wikipedia category, the goal of our system is to decide whether it warrants its own article in Wikipedia.

For a given input named entity related to a particular Wikipedia category, we first extract the training data.

1. named entities - $e_1, e_2, ...e_N$ related to that category with Wikipedia articles $A_1, A_2, ...A_N$

2. named entities - $e'_1, e'_2, ...e'_M$ related to that category which do not have Wikipedia pages

For collecting named entities that are having Wikipedia articles, we can use the list of all named entities present in the Wikipedia category. For example, we can get the list of names of Indian actors that have Wikipedia pages through the page: https://en.wikipedia.org/wiki/Category:Indian_film_actors by recursively crawling through its sub-categories.

However collecting named entities that are not having Wikipedia articles is not as straight-forward. We need to crawl articles where we can find such named entities and after extracting them we need to filter the list of named entities which do not have their entry in Wikipedia. For example, we can get the list of Indian actors who do not have Wikipedia pages by crawling the content under the "Cast" sub-section in the articles belonging to the Wikipedia category https://en.wikipedia.org/wiki/Category:Indian_films. This content gives the list of all actors, from which we filter the actors who are not included in Indian Actors category. We could also use websites like *imdb.com* to extract actors that are not included in Wikipedia.

However, using named entities whose wikipedia articles got deleted from wikipedia for our negative sample would be more appropriate. Considering our category-centric approach, collecting deleted articles specific to a particular wikipedia category is challenging. We would work on the same as a part of our future work for building a stronger training set.

In the training data, named entities $e_1, e_2, ...e_N$ are labelled as Notable entities and $e'_1, e'_2, ...e'_M$ are labelled as Not notable entities. Using this training data set, we build a Boolean classification model (Section 3.2) with information related to the named entity in the web, reliability of that content and the significant coverage of that named entity in the content as the major features for the classifier. For a given test named entity as input, this classification model returns a class label with the decision about its notability. In further sections, we discuss details about the features and the classification model.

3.1 Features

We evaluate a set of features which determine the notability of a named entity. According to Wikipedia's official guidelines, A topic is presumed to merit an article if:

1. It meets either the general notability guideline, or the criteria outlined in a subject-specific guideline.

2. It is not excluded under the "What Wikipedia is not policy"[6].

We determine an estimation of each guideline except for the subject specific notability as it contains complex rules, which we did not focus in this paper. First, general notability guidelines can be broadly summarized in two rules mentioned below.

- *Reliability of the information collected*: Information about the named entity has to present in web sources which are considered reliable.

- *Significant coverage of the named entity*: There has to be significant coverage of the named entity in such web articles from which the content is taken.

Considering the above mentioned rules, we use information collected from the web for the named entity as a resource to

[6] en.wikipedia.org/wiki/Wikipedia:What_Wikipedia_is_not

Table 1: Reliable domain Features for Indian Actors Wikipedia Category

External reference domains		**Frequent domains**		**Final chosen features**
el_i	$F_{el}(el_i)$	rd_j	$F_{rd}(rd_j)$	
imdb.com	2126	imdb.html	1328	imdb.com
en.msidb.org	133	filmibeat.com	993	bollywoodhungama.com
bollywoodhungama.com	75	bollywoodlife.com	665	en.msidb.org
gomolo.com	25	moviesdosthana.com	436	filmibeat.com
filmibeat.com	11	filmyfolks.com	265	bollywoodlife.com
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

check its notability. We discuss more details about this in Sections 3.1.1 and 3.1.2.

"What Wikipedia is not" provides rules that mostly apply to the information contained in the article.

Some such rules include

- Article should not have editorial bias (No original opinions / facts discovered in news editorials)
- Article should not include non-verifiable content (websites have to be accessible by everyone)
- Article should not contain content included in social networking websites
- Article should not contain promotional content

So, while collecting the information for a named entity, we include content that abides by these rules by excluding the well known web-domains that belong to such categories.

3.1.1 Reliable Domain features

For a particular Wikipedia category, we define a set of reliable web-domains as the important features. To the best of our knowledge, there has been no research done on automatically identifying the reliable web domains. The methods that we investigated for getting such reliable domains are mentioned below.

- **External Reference domains**: As discussed in [5], web-domains cited in the external references section can be leveraged for determining the reliable domains because such domains harvest the wisdom of the reviewers and editors who abide by the Wikipedia guidelines. We collect all articles belonging to the Wikipedia category of the named entity chosen and parse them to get the content under External references section. We extract frequent domains from such content. Let el_1, el_2, el_3, ... el_p be the list of external reference domains extracted and F_{el} be a function such that

  $F_{el}(el_i)$ = frequency of el_i in articles of the Wikipedia category where $i \in [1, p]$

- **Frequent domains**: Here we leverage the kind of web domains that are frequently seen across web for named entities that have Wikipedia pages. For each of the named entity of that Wikipedia category, we collect the search results from the training data for a query: "Named Entity"+ "Wikipedia Category". We store the web domains of each of the web-links in the results. Let rd_1, rd_2, rd_3, ...rd_q be the list of domains extracted and F_{rd} be a function such that

  $F_{rd}(rd_j)$ = frequency of rd_j in the domains obtained from performing a Google search for named entities of the Wikipedia category where $j \in [1, q]$

We choose reliable domain features RF_k across el_is and rd_js such that

$$\alpha * F_{el}(RF_k) + \beta * F_{rd}(RF_k) \geq t \text{ where } F_{rd}(RF_k) > 0 \quad (1)$$

External reference domains have to be given more significance as they are officially authorised by the Wikipedia reviewers. So, weighted parameters p and q for External reference domains and Frequent domains are chosen such that $\alpha > \beta$. Threshold parameter t depends on the number of dimensions we wish to choose. Of the extracted web domains RF_k, we exclude domains which do not stand by the **"What Wikipedia is not"** guidelines. In particular, we remove well known webdomains belonging to social networking, news, and non-textual categories. Some such hardcoded webdomains in our approach include facebook.com, twitter.com, youtube.com, instagram.com and domains that have term "blog" in their urls. Table 1 shows the external reference domains, frequent domains and the reliable domain features chosen for Wikipedia Category Indian film actors.

For our classification model, the domains RF_k become the dimensions and the instance values for the vectors would be 1 or 0 which correspond to presence or absence of that web-domain article for that particular named entity in web. Table 3 shows a sample Vector space representation of our training dataset using only reliable domain features as dimensions.

Table 3: K-dimension Vector Model for training data

Vector	RF_1	RF_2	....	RF_K	**ClassLabel**
Entity-1	1	0	...	0	Yes
Entity-2	1	1	...	0	No
Entity-3	0	0	...	0	Yes
.	.	.	...	.	.
.	.	.	...	.	.
Entity-N	0	0	1..	0	No

Table 2: Entity Salience Measures

Entity Salience Feature	**Description**	**Value (Scaled from 0 to 10)**
Positional Index (E_p)	check on occurrence of named entity or its mention in first sentence	$E_p = 10$ if entity occurs in first sentence and 0 otherwise
First Location (E_f)	Index of the sentence in which the entity was mentioned for the first time.	$E_f = 10$ - (sentence number of first location*10/total number of sentences)
Initial head count (E_i)	Number of times the entity occurred in first three sentences of the article.	$E_i =$ (number of occurences in first three sentences*10/3)
Head-count (E_h)	Number of times the entity occurs in the article	$E_h =$ total number of occurences / number of sentences

3.1.2 Entity Salience Features (Significant Coverage)

We use entity salience of named entity defined in [2] as a measure for its significant coverage in an article. Entity salience is a relevance score given to each entity in a document. Some of the entity salience measures that we use as features are mentioned in Table 2.

Let us consider a web-domain RF_k obtained from Sec 3.1.1. For a named entity, we get the article A_k in that web-domain. First, we use boiler pipe [3] to extract the textual content from the article. We resolve the co-references of the entity mentions using the Stanford Co-reference resolution tool [4]. From this content, we compute the entity scores (E_p, E_f, E_i, E_h) mentioned in Table 2. These scores are with respect to the web-domain RF_k, similarly we can compute the entity scores for all other domains chosen in Section 3.1.1.

Table 4 shows a part of the vector model with domain RF_k and the entity salience scores for that domain as domains. The value corresponding to the entity salience feature in the vector would be its entity salience value (scaled from 0 to 10) of the named entity, if there exists an article in the domain considered and 0 otherwise. Such vector model built over all the entities in the training dataset is used for the classifier we choose.

Table 4: Dimensional Vector Model with a single domain RF_k and its entity salience scores as features

Vector	RF_k	$E_p(RF_k)$	$E_f(RF_k)$	....	**Label**
Entity-1	1	8	9	...	Yes
Entity-2	1	0	0	...	No
Entity-3	0	0	1	...	Yes
.	.	.	.	...	.
.	.	.	.	...	.
Entity-N	0	0	0	...	No

3.2 Classification Model

The domains and the entity salience scores of the named entity in the corresponding articles as shown in Table 3 and 4. We merge both the features to get the vector model similar to Table 4 but for all the N reliable domains that we choose, instead of one reliable domain feature. Our final vector model would have dimensions: RF_1, $E_p(RF_1)$, $E_f(RF_1)$, $E_i(RF_1)$, $E_h(RF_1)$, RF_2, $E_p(RF_2)$, $E_f(RF_2)$, $E_i(RF_2)$, ... RF_N, $E_p(RF_N)$, $E_f(RF_N)$, $E_i(RF_N)$, $E_h(RF_N)$. The vectors formed for entities would have a value 0 or 1 for reliable domain feature dimensions and a value ranging from 0 to 10 for Entity salience feature dimensions.

Table 5: Cross Validation results for different classifiers (entity salience factors included for each)

Classifier	**Precision**	**Recall**	**F-score**
Naive Bayes	0.865	0.847	0.845
Logistic Regression	0.923	0.921	0.921
Decision Table	0.901	0.897	0.896
Random Tree	0.910	0.908	0.908
SVM	0.925	0.923	0.923

We investigate the dimensions required and give the vectors formed for the named entities to train the classifier model which has the class labels Yes (Notable Entity) or No (Not Notable Entity). This model can be used for classifying any new named entity in the same Wikipedia category.

For any numeric continuous data, classifiers like logistic regression and SVM would be good techniques to choose. However, we tested across various classifiers and chose the best suitable for our data. Table 5 shows the 10 fold cross validation accuracies of various classifiers over the training dataset on Indian Actors. We see that SVM classifier outperforms the other major classifiers and hence we choose it for our model.

4. EXPERIMENTS AND RESULTS

4.1 Dataset

We collected a dataset for Wikipedia Category: Indian Actors containing of 1000 actors who do not have Wikipedia articles and 1000 actors who have Wikipedia articles.

For actors who have Wikipedia pages, we collected the names the actors by recursively crawling the page: https://en.wikipedia.org/wiki/Category:Indian_film_actors. For actors who do not have Wikipedia pages, we crawled the articles in wiki category: https://en.wikipedia.org/wiki/Category:Indian_films. From the Cast section (which usually contains the list of actor names in the film) of the articles, we have extracted the names of actors who are not hyperlinked with their corresponding Wikipedia article and there after cross-verified their non-existence in Wikipedia.

For each of these actors, we got the google search results (along with their domains) for query :"Actor Name" + Indian film actor. We also did a focused search for first ten frequent domains chosen. For instance, we used IMDbPY

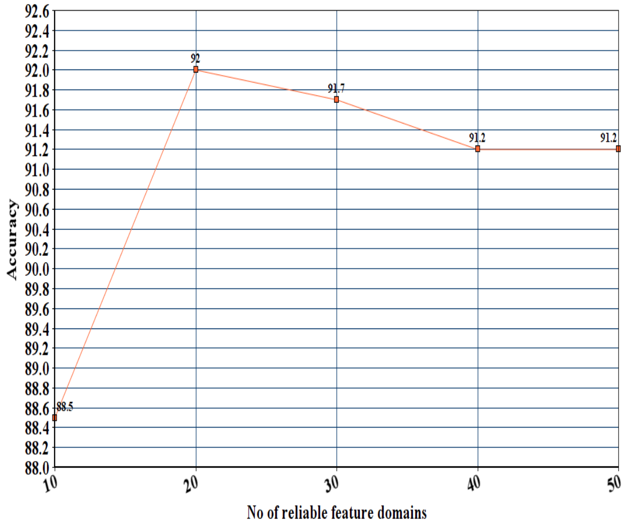

Figure 1: Cross Validation results based on dimension count of Reliable domains (entity salience factors included)

[1] to retrieve and manage the data from imdb.com. We use this dataset and the web results to evaluate our system.

4.2 Results

We examine the role which the number of dimensions and the features included in classifier model play for evaluating our model. Figure 1 shows the 10-fold cross validation accuracy scores across the classifier models with varying number of dimensions (reliable domain features). We have included all of their entity salience scores as well. The X axis on the graph indicates the number of top reliable domain features chosen from their list sorted in the decreasing order of the $\alpha * F_{el}(RF_k) + \beta * F_{rd}(RF_k)$ score. We see that the accuracy gradually increases till 20 and then decreases showing that the less weighted reliable domains do not capture the notability efficiently.

We use 20 reliable domain features and see which entity salience features play a major role in the classification model. Table 6 gives the Precision, Recall and F-Score values for various combinations of the entity salience features.

Considering only reliable domain features which mention about the existence of the article in a domain will lead us to possible junk articles and articles that are irrelevant to the named entity. Hence we see that including only reliable domain features gave significantly lower accuracies compared to the ones which have entity salience scores included along with the reliable domain features.

Table 6: Accuracies across various combinations of the entity salience features

Feature	Precision	Recall	F-score
Only Reliable domains	0.844	0.840	0.839
+ Positional Index	0.899	0.896	0.896
+ Head-count	0.909	0.907	0.906
+ Initial head count	0.905	0.904	0.904
+ First Location	0.926	0.924	0.924
+ Head-count + Initial head-count + First Location + Positional Index	0.925	0.923	0.923

Table 7: Results for a few actors by our system

Notable actors who do not have wikipedia pages	Not-Notable Actors having Wikipedia pages
Ruchita Prasad	Som Nath Sadhu
Rohit Raj Goyal	Nikhil Wairagar
Dilip Thadeshwar	Hamom Sadananda
Arun kadam	Bonium Thokchom
Jatin Grewal	Gokul Athokpam

An entity salience feature working better depends on how well it could capture the significance (coverage) of the entity, which means higher score score for entity that has greater significance in the article and lower score for entities who are not talked much about in the article. In Table 6, we see that entity salience score: First Location individually works marginally better than the other entity salience scores. We discuss what could have caused such marginal difference by considering values which the various entity salience features in the Table 2 give in various scenarios.

Below quoted is the information included in the imdb article of Sunil Dutt[7].

"Actor, social activist and politician. Sunil Dutt wore many hats and excelled in a plethora of roles that came his way - both on and off screen. Born on June 6th, 1929, Sunil Dutt grew up... so on"

This content actually talks about Sunil Dutt, so ideally the entity salience score should be high. We see that Positional index gives a score 0 as there is no mention about Sunil Dutt in the first sentence. If we consider a similar article as above where first three lines do not have the mention about the article but the sentences following that have information related to the actor, Initial head count would also have been 0. Additionally, if the content is very large with information about the named entity included only in first few paragraphs, then Head-count would also be relatively low. However first location score would still be high in all such cases as the mention about Sunil Dutt is still there in the first few sentences.

The model built over our training data can used for determining the notability any new actor. Table 7 shows the some actors who do not have wikipedia pages, but were marked as notable by our system, the reason being they having content in reliable frequent domains we chose. Adding wikipedia pages for such notable entities automatically would lead us towards the automation ensemble that we discussed in Section 1. We also show in table 7, some actors who have wikipedia pages but were classified as not-notable because they did not have content in our chosen reliable domains like imdb.com. Such wikipedia pages can be sent to the

[7] http://www.imdb.com/name/nm0004570/bio?ref_=nm_ov_bio_sm

reviewers for “Article for Deletion” discussions for further investigation.

5. CONCLUSION

In this paper, we implemented a system that automatically evaluates the decision related to the notability of a named entity which in turn warrants its article inclusion in Wikipedia. Our early efforts in this direction show that reliable domains and entity salience features can be good measures to determine the notability of a named entity.

However this problem paves a path for various directions of future research. Our method is applicable for categories like actors, movies or software companies as they have common reliable web-domains across the named entities belonging to the category. However, the categories that have articles with concepts like temperature, physical phenomena, etc require more sophisticated approaches for automating the reliability and significant coverage of the named entities. Automation of features like subject-specific notability which involve more complex rules and more efficient reliability automation can lead us towards automating notability of any kind of entry in Wikipedia. As discussed in section 2 we can also extend our notability features to support cross-language content.

In our future work, we would focus on the aforementioned techniques which would make notability more scalable unlike our current approach that restricts to named entities.

6. REFERENCES

[1] D. Alberani. Python package: Imdbpy. https://github.com/alberanid/imdbpy.

[2] D. Gillick and J. Dunietz. A new entity salience task with millions of training examples. In *Proceedings of the European Association for Computational Linguistics*, 2014.

[3] C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM ’10, pages 441–450, New York, NY, USA, 2010. ACM.

[4] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[5] Y. Pochampally, K. Karlapalem, and N. Yarrabelly. Semi-supervised automatic generation of wikipedia articles for named entities. In *Wiki, Papers from the 2016 ICWSM Workshop, Cologne, Germany, May 17, 2016*, 2016.

[6] C. Sauper and R. Barzilay. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL ’09, pages 208–216, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[7] E. Wulczyn, R. West, L. Zia, and J. Leskovec. Growing wikipedia across languages via recommendation. *CoRR*, abs/1604.03235, 2016.

[8] C. Yao, S. Feng, X. Jia, F. Zhou, S. Shou, and H. Liu. Autopedia: Automatic domain-independent wikipedia article generation, 2011.