



MASTER IN COMPUTING

MASTER THESIS

# Enhancing Quality Arabic Web Content through Multilingual Information Retrieval Methods

*Abdulfattah Safa*

supervised by

*Dr. Adnan Yahya*

*This Thesis was submitted in partial fulfilment of the requirements for the Master's  
Degree in Computing from the Faculty of Graduate Studies at Birzeit University,  
Palestine*

*May, 2019*



# Enhancing Quality Arabic Web Content through Multilingual Information Retrieval Methods

By Abdulfattah R. Safa

Approved by the thesis committee:

**Dr. Adnan Yahya, Birzeit University**

---

**Dr. Radi Jarrar, Birzeit University**

---

**Dr. Mustafa Jarrar, Birzeit University**

---

## **Abstract**

Access to quality web content is an integral part of modern life. While Arabic content is increasing rapidly, it is still far from adequate in terms of quantity, quality and timeliness and is well below what is available in many other languages. To meet user needs, it helps to give Arabic users access to foreign web content. The focus of the thesis is to grant Arabic users access to quality web content in other languages to meet their information needs. This includes access to: textual data, structured databases and annotated multimedia elements. For that we rely on Cross Lingual Information Retrieval (CLIR) concepts such as cross lingual document similarity, relevance measures, named entity recognition/transliteration, document quality assessment tools and knowledge extraction results [1,2]. Among the approaches we build on are Semantic Association (Explicit and Latent) [1,3,4], Knowledge Extraction [5,6,7] and Machine Translation and Transliteration [8]. The methods developed will be utilized to increase the Arabic web content, to suggest material for further processing using crowd sourcing and, as a byproduct, for cross lingual plagiarism detection.

### **Acknowledgements**

I would first like to thank my thesis advisor Dr. Adnan Yahya for the patience guidance, encouragement and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who responded to my questions and queries so promptly. His positive outlook and confidence in my research inspired me and gave me confidence.

I also would like to express my very profound gratitude to my parents and to my wife Raghad for providing me with support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Thank you  
Abdulfattah.

# Contents

<b>Abbreviations</b>	<b>10</b>
<b>1 Introduction</b>	<b>11</b>
1.1 Introduction . . . . .	11
1.2 Importance and Expected Impact . . . . .	12
1.3 Research Problem and Goals . . . . .	12
1.4 Research Methodology . . . . .	13
1.4.1 Cross Lingual Document Similarity . . . . .	13
1.4.2 Access to Foreign Language Structured Data . . . . .	14
1.4.3 Access to Foreign Language Multimedia Elements . . . . .	15
<b>2 Literature Review</b>	<b>17</b>
2.1 Statistical or Corpus-based Similarity . . . . .	17
2.2 Knowledge Base-based Similarity . . . . .	19
2.3 Lexico-Syntactic Patterns-based Similarity . . . . .	21
2.4 Explicit Semantic Analysis . . . . .	22
<b>3 Proposed Approach</b>	<b>27</b>
3.1 Cross Lingual Document Similarity . . . . .	27
3.2 Access to Foreign Language Structured Data . . . . .	51
3.2.1 Query Foreign Language Structured Data . . . . .	51
3.2.2 Generate Arabic Structured Data from Foreign Language Structured Data . . . . .	60
3.3 Expanding Work to Other Languages . . . . .	78
<b>4 Results</b>	<b>79</b>
4.1 Cross Lingual Document Similarity . . . . .	79
4.1.1 Selecting Representative Category . . . . .	79
4.1.2 Wikipedia Featured Articles . . . . .	79
4.1.3 Translated News Articles . . . . .	82
4.1.4 Bilingual Tweets . . . . .	83

4.1.5	Global Voices Parallel Corpus . . . . .	84
4.1.6	Using Part of Speech Tagging . . . . .	97
4.1.7	Using Category Links . . . . .	101
4.1.8	Vector Length . . . . .	106
4.2	YAGO Facts Translation . . . . .	116
4.2.1	Fact Translation Status . . . . .	116
4.2.2	Fact Translation Quality: Human Ranking . . . . .	119
4.2.3	Fact Translation Quality: Compare to Context Search Engine Results . . . . .	119
<b>5</b>	<b>Conclusion and Future Work</b>	<b>121</b>
	<b>Appendices</b>	<b>129</b>
<b>A</b>	<b>Cassandra Create Script</b>	<b>130</b>
<b>B</b>	<b>Wikipedia Featured Articles</b>	<b>134</b>
<b>C</b>	<b>Translated News Articles</b>	<b>137</b>
<b>D</b>	<b>Bilingual Tweets</b>	<b>141</b>

# List of Figures

2.1	ESA Semantic Interpreter[9]	23
3.1	Spark Cluster Overview	40
3.2	Building Weighted Inverted Index for Documents	45
3.3	YAGO Spotlx Interface	52
3.4	Birzeit University Wikidata Page	55
3.5	DBpedia Extraction Framework[10]	56
3.6	University Class in DBpedia	57
4.1	Percentage of Documents with Rank $\leq 10$ to the Total number of Documents in the Collection for Wikipedia Featured Articles	85
4.2	Percentage of Documents with Rank $\leq 10$ to the Total number of Documents in the Collection for Translated News Articles	86
4.3	Percentage of Documents with Rank $\leq 10$ to the Total number of Documents in the Collection for Bilingual Tweets	87
4.4	Percentage of Documents with Rank $\leq 10$ to the Total number of Documents in the Collection for Global Voices	88
4.5	Percentage of Documents with Rank $\leq 10$ to the Total number of Documents in the Collection for Wikipedia Featured Articles	89
4.6	Percentage of Documents with Rank $\leq 5$ to the Total number of Documents in the Collection for Translated News Articles	90
4.7	Percentage of Documents with Rank $\leq 5$ to the Total number of Documents in the Collection for Bilingual Tweets	91
4.8	Percentage of Documents with Rank $\leq 5$ to the Total number of Documents in the Collection for Global Voices	92
4.9	Percentage of Documents with Rank = 1 to the Total number of Documents in the Collection for Wikipedia Featured Articles	93
4.10	Percentage of Documents with Rank = 1 to the Total number of Documents in the Collection for Translated News Articles	94
4.11	Percentage of Documents with Rank = 1 to the Total number of Documents in the Collection for Bilingual Tweets	95

4.12	Percentage of Documents with Rank = 1 to the Total number of Documents in the Collection for Global Voices . . . . .	96
4.13	Percentage of Documents with Rank $\leq 10$ to the Total number of Documents in the Collection with and without Applying Part Of Speach (POS) Tagger . . . . .	99
4.14	Percentage of Documents with Rank $\leq 5$ to the Total number of Documents in the Collection with and without Applying POS Tagger	100
4.15	Percentage of Documents with Rank = 1 to the Total number of Documents in the Collection with and without Applying POS Tagger	101
4.16	Percentage of Documents with Rank $\leq 10$ to the Total number of Documents in the Collection . . . . .	104
4.17	Percentage of Documents with Rank $\leq 5$ to the Total number of Documents in the Collection . . . . .	105
4.18	Percentage of Documents with Rank Equals to 1 to the Total number of Documents in the Collection . . . . .	106
4.19	Similarity Test Scores when Using 47012 Categories . . . . .	108
4.20	Similarity Test Scores when Using 24825 Categories . . . . .	109
4.21	Similarity Test Scores when Using 19031 Categories . . . . .	110
4.22	Similarity Test Scores when Using 13786 Categories . . . . .	111
4.23	Similarity Test Scores when Using 12076 Categories . . . . .	112
4.24	Similarity Test Scores when Using 9178 Categories . . . . .	113
4.25	A Snapshot of the Actual Score Analysis . . . . .	114
4.26	Similarity Test Aggregate Score . . . . .	115



# List of Tables

2.1	Top ranked Wikipedia Articles in English and German (mapped to English) for query “Scary Movies” . . . . .	25
2.2	The positions of some Wikipedia articles in the ranked vectors in English and German (mapped to English) for query “Scary Movies”	25
3.1	Total Number of Retrieved Wikipedia Pages and Categories in Arabic and English . . . . .	36
3.2	Statistics about the Stored Data . . . . .	42
3.3	Representative Categories Word Count . . . . .	44
3.4	Representative Categories . . . . .	44
3.5	POS Tags of the Terms Removed from Documents . . . . .	47
3.6	Birzeit University Entity Relations in YAGO . . . . .	53
3.7	Birzeit University Item Properties in Wikidata . . . . .	54
3.8	Birzeit University Item Identifiers in Wikidata . . . . .	54
3.9	Birzeit University Entity Properties in DBpedia . . . . .	58
3.10	Arabic Mapping of Classes and Properties in DBpedia . . . . .	58
3.11	A Comparison between YAGO, Wikidata and DBpedia . . . . .	59
4.1	Similarity Test Results for Queries of Arabic Articles against Collections of Different Types and Languages . . . . .	80
4.2	Similarity Test Results for Queries of English Articles and Documents of Different Collections . . . . .	81
4.3	Similarity Test Results for Queries of Arabic Articles Introductions and a Collection of English Article Introductions . . . . .	82
4.4	Similarity Test Results for Translated News Articles . . . . .	83
4.5	Similarity Test Results of Bilingual Tweets . . . . .	83
4.6	Similarity Test Results of Global Voices Parallel Corpus . . . . .	84
4.7	Similarity Test Results for Queries of Arabic Articles and Documents of Different Collection with and without Applying POS Tagger	97
4.8	Similarity Test Results for Queries of English Articles and Documents of Different Collection with and without Applying POS Tagger	97

4.9	Similarity Test Results for Arabic Article Introductions and English Article Introductions with and without Applying POS Tagger . . .	98
4.10	Similarity Test Results for Tweets with and without Applying POS Tagger . . . . .	98
4.11	Similarity Test Results for Queries of Arabic Articles and Documents of Different Collection with and without Using Category Links	102
4.12	Similarity Test Results for Queries of English Articles and Documents of Different Collection with and without Using Category Links	102
4.13	Similarity Test Results for Arabic Article Introductions and English Article Introductions with and without Using Category Links . . . .	103
4.14	Similarity Test Results for Translated News Articles with and without Using Category Links . . . . .	103
4.15	Similarity Test Results for Tweets with and without Using Category Links . . . . .	103
4.16	Similarity Test Results of Global Voices Parallel Corpus with and without Using Category Links . . . . .	104
4.17	Translated YAGO Fact Subject Translation Status per Property . .	117
4.18	Translated YAGO Facts Objects Translation Status per Property .	118
4.19	Translated YAGO Facts Human Ranking Summarization . . . . .	119
4.20	Translated YAGO Facts Compared to Context Search Engine Results	120

# List of Algorithms

1	Translate Wiki Title . . . . .	63
2	Translate Wikipedia Article Title . . . . .	64
3	Translate Wikidata Title . . . . .	65
4	Translate Wikidata Item Label . . . . .	66
5	Translate Wikidata Item Alias . . . . .	67
6	Translate Wikidata Item Description . . . . .	68
7	Translate YAGO Subject . . . . .	69
8	Translate YAGO Object . . . . .	70
9	Translate YAGO Object Using Wiki . . . . .	75
10	Translate YAGO Object Using Google Translate API . . . . .	76

# Abbreviations

**CL-ESA** Cross Lingual Explicit Semantic Association

**CLIR** Cross Lingual Information Retrieval

**ESA** Explicit Semantic Analysis

**IR** Information Retrieval

**LCS** Least Common Subsumer

**LSA** Latent Semantic Analysis

**NGD** Normalized Google Distance

**POS** Part Of Speech

**RDF** Resource Description Framework

**SVD** Singular Value Decomposition

**TF-IDF** Term Frequency - Inverse Document Frequency

# Chapter 1

## Introduction

### 1.1 Introduction

Reliance on web content for meeting peoples information needs has been growing rapidly [11]. Most people take to the Internet for their information needs. Information Retrieval (IR) is becoming an integral part of our lives. In many locations the vast majority of Internet users do searches several times a day and that proportion has been increasing rapidly. We believe that the coverage and reliability of information available on the web are major contributors to the heavier reliance of users on web content. The proportion of Arab users of the Internet (around 42.69%) is well below that of more developed OECD countries (78.68%), and even lower than the world average (45.79%)<sup>1</sup>. The volume of Arabic web content has been increasing rapidly. However, it is still below what is available in other languages when the number of speakers is factored in. Another issue is that of the quality of Arabic web content in terms of coverage, style and timeliness. Studies show that the quality is well below what is available in other languages. Given that we face both coverage and quality problems when dealing with Arabic web content, the issue of giving Arabic speaking users access to quality data in other languages becomes a possible solution pending efforts at increasing quality Arabic web content.

Arabic web content creation is gaining interest in the Arab World with many initiatives directed at creating new content [12, 13, 14, 15, 16]. Arabic content retrieval is a focus of many researchers, mostly outside the Arab region and the contribution of researchers in the Arab Region is well below what is desired [5, 17].

The main interest is in interpreting existing content, formal and in social media, in many cases for political and security reasons [13, 18]. In this sense our work will be pioneering in that it has its main emphasis on the interaction between content

---

<sup>1</sup> <http://data.worldbank.org/topic/infrastructure> (2017 UN statistics)

generation and content quality assessment and retrieval.

This thesis can be viewed as part of CLIR. CLIR is the branch of IR where the user information need (query) is expressed in a language different from the language of the information retrieved. This is a maturing discipline<sup>2</sup>, though we believe not enough work is being done where one of the language pairs is Arabic [2, 19]. While we concentrate on Arabic-English CLIR, many of the concepts addressed here are applicable to other language pairs.

## 1.2 Importance and Expected Impact

The rapid increase in knowledge generation, most of which is electronic, renders traditional methods for information access/digestion impractical. The discrepancy in the amounts of data generation in different languages is clear and may be increasing. This creates another manifestation of the haves and have-nots in this case in terms of timely access to quality knowledge. The Arab user is disadvantaged by the relative size of Arabic content and the state of Information Retrieval tools available to access existing data.

Access to larger amounts of data from the web is a major enabler in a knowledge based economy. Limiting access to Arabic content will maintain the status quo disadvantages Arabic users. Thus, timely access to content in other languages, even by users not versed in these languages, acquires added importance. The thesis seeks to simplify and facilitate this access and therefore we believe that the impact will be far reaching, with implications for education, industry, economy, culture and other aspects of our lives.

## 1.3 Research Problem and Goals

**Arabic vs. Non-Arabic Web Content:** Much of the material on the web is in languages other than Arabic and may not be available to Arab speaking users. Included here useful data in the form of text, video and audio. Major parts of this material can be of interest to Arab surfers as it may describe entities/services they need, such as How To videos, info text articles (say Wikipedia), topical and general data/knowledge bases, product manuals and product evaluations, health and safety related information, but not readily available in Arabic. Making this material accessible to Arabic users even in a restricted manner, will enhance the usefulness of the Web and may even encourage better connectivity by increasing the returns on investment.

---

<sup>2</sup>The first CLIR workshop was organized in 1996.

[http://en.wikipedia.org/wiki/Cross-language\\_information\\_retrieval](http://en.wikipedia.org/wiki/Cross-language_information_retrieval)

To get an idea about the relative sizes of Arabic and Non-Arabic content, suffice it to say that the estimates say [15] Arabic Web is about 2% of the total content<sup>3</sup>, and the article count in the Arabic Wikipedia is only about 14% of that of English and 22% of the Swedish (816,836 vs 5,860,179 vs 3,747,559, respectively<sup>4</sup>). If one factors in quality, the picture becomes gloomier[16,18]. The need for information is here and now. One cannot wait to get access to available data, which is there in other languages. There is a substantial need for a short-cut access to this data for Arabic speakers. The goal of this thesis is to grant Arabic speakers fast access to high quality material available in other languages. For that, We will rely on concepts from CLIR.

Our goal is to make the user aware of the existence of the quality material in other languages, and based on the perceived utility (say through a ranking system) one may resort to crowdsourcing to have the material annotated/translated.

## 1.4 Research Methodology

The main focus of the thesis is multilingual content accessibility in Arabic. Granting access to non-Arabic content to Arabic speaking users will be done using a multiplicity of methods to account for the various types of data one needs to access. The following aspects of access to foreign data are the most important, and they will be the focus of the thesis, each utilizing the proper tools for the task at hand:

### 1.4.1 Cross Lingual Document Similarity

Given an Arabic Information need (query): Designate the relevant documents both in Arabic and the Foreign languages. The first part is a standard search task and will not be the focus of the current thesis. For this we will rely on existing tools, including enhancements[20,21]. Estimating relevance for cross-lingual documents (say an Arabic Query and an English document) is not a straightforward operation and needs extensive research. More so if we are talking about semantic, and not only syntactic similarity. That is, if we are interested in documents conveying similar meaning not necessarily using the same set of words. For that we need tools to detect and quantify similarities between Arabic and English documents and use this similarity to rank foreign language results. Since our basic premise will be that the number of Arabic documents is small, we will limit our consideration to the first N most relevant documents, where N is a number between 3 and 10 but not exceeding 50 percent of returned results, for efficiency purposes.

---

<sup>3</sup> <http://www.worldwidewebsite.com/> (December 2014)

<sup>4</sup> [http://meta.wikimedia.org/wiki/List\\_of\\_Wikipedias](http://meta.wikimedia.org/wiki/List_of_Wikipedias) (May 2019)

The main need is to make the relevant foreign language documents available to Arabic speakers. For that we will use statistical analysis of user needs to suggest translation candidates for an essentially crowd sourcing tasks. Machine translation into Arabic may be provided as an initial interpretation of the relevant documents, and may also serve as a basis for future improvement.

We may use the foreign language content as a sort of relevance feedback to improve the search for Arabic documents. Also we may use the search results in Arabic (or part of that) to improve the search results for the query in the foreign language.

The tools employed here will be based of Cross Lingual Explicit Semantic Association (CL-ESA) modified to work with concepts that are language independent in the presence of surface translation and semantic tools such as categories as opposed to the usual textual data in the form of Wikipedia Articles and categorized directories [1, 3, 4, 22].

### 1.4.2 Access to Foreign Language Structured Data

Search in unstructured data (say free text) is usually imprecise, slow and is unable to deal with complex queries. It is not easy to tell a Google style search engine to return a list of leaders who headed three given countries with overlapping terms in office. Such queries are easily answered in a database with countries, leaders and their service dates. More people are interested in asking questions rather than searching for keywords. Given that most of web content has the form of unstructured data (free text) an issue is how to convert free text to database tuples: that is how to extract knowledge expressible in a database from free text. Much work has been done in this area of knowledge Extraction resulting in huge tables with tuples that can be queried by user s[6, 21].

One maybe interested in one important aspect of this approach, which is to grant Arabic speakers access to tables with knowledge extracted from texts in foreign languages. We believe that by translating a limited number of attributes and a transliteration of a large number of entities we can gain access to a wealth of knowledge that may not be available otherwise. We need not be limited to the results of knowledge extraction but may extend that to useful databases in a multiplicity of fields: consumer protection data with product reviews, health related data with diseases, symptoms and the likes and geographical data and much more. The challenge here is the translation of the large amounts of data, which has two components:

1. Translation relation names and predicates: the amount of this data is small and covers large databases. Even manual translation can be applied here, with the possibility of crowdsourcing. This component is to be treated as a



high priority task and the translation needs to be of high quality.

2. Translating entity names: person names, location names, company names, and similar. To be scalable one needs to use machine translation/transliteration tools for this purpose [8]. Crowdsourcing can be used to refine/correct machine generated transliterations. Due to its size, this is a lower priority component and one may want to order the candidate material based on demand and perceived importance. For some cases one may even opt to provide tuples/lists with only attribute names translated. For example, in response to the query list the last three Secretary Generals of the UN and their service time and Countries of Origin will be a table with Arabic Predicates “الأمين العام” and “خدم من سنة” and “البلد” and “خدم إلى سنة” and English (or mixed) data {Ban Ki-Moon, Kofi Anan, Boutros Ghali} and their service dates and country of origin as follows:

الأمين العام	خدم من سنة	خدم إلى سنة	البلد
Ban Ki-moon	2007	2016	Korea
Kofi Anan	1997	2006	Ghana
Boutros Ghali	1992	1996	Egypt

The tools used here will be based on processing structured data extracted into tables from textual materials in Arabic and foreign languages [5, 7, 16]

### 1.4.3 Access to Foreign Language Multimedia Elements

This includes pictures and video. The positive side here is that much of the material may be understandable even in the absence of translation. What may be required for better understanding is dubbing/subtitling and/or translating meta-data. However, identifying such content may be less of a problem as the amount of text is usually small and is characterized by a limited number of terms clearly identifying the content. This suggests using a form of term translation to identify content in foreign language in response to an Arabic query. So for multilemdia elements we have two stages:

1. To find the material relevant to the query. To us this will take the form of matching the terms of the query with the meta-data of the multilemdia object. The found multilemdia elements will be ranked based on the (cross-lingual) similarity of the query and the multilemdia object meta data
2. Once the object is found translating it into Arabic may become a problem. One can use Machine translation for captions and shorter metadata. For audio material, much more sophisticated methods are needed. Machines

can help little here and we will resort to crowd sourcing tools to translate materials based on perceived importance/need

Here we will use in-house methods building on machine learning tools and manual annotations. We will also design a prototype system to simplify crowdsourcing for this effort.

One can view the proposed work as adapting existing techniques to the Arabic setting in a manner that accounts for the peculiarities of Arabic and putting all that to a practical use: granting better access to quality content to Arabic speakers without burdening the user with learning another language. Awareness of the existence of such content may encourage the user to explore more and may encourage the creation of Arabic content through translation. The approach can also be used to assess cross language document similarity, something that may help detect cross-language plagiarism.

# Chapter 2

## Literature Review

The problem of measuring the semantic similarity between two texts has been studied deeply in literature due to its roles in many applications, such as information retrieval, relation extraction, text summarization and similarity and error correction.

Semantic similarity approaches are classified based on different criteria. Gutpa and Kumar [23] classified the semantic similarity approaches into corpus and knowledge base-based approaches. Whereas Majumder [24] called the knowledge base based approaches as topological/knowledge approaches and the corpus based approaches as statistical/corpus approaches then added a new class: string-based approaches. Gomaa and Fahmy [25] added hybrid approaches a fourth class.

In this chapter we present a technical background about the some of the work done in this field.

### 2.1 Statistical or Corpus-based Similarity

To measure semantic similarity between words, information related to these words can be extracted from large corpora then metrics can be applied on it. So statistical similarity is learned from data. Corley, Mihalcea and Strapparava [26] suggested to use two metrics: pointwise mutual information [27] and latent semantic analysis [28]. Another important metric also will be presented, which is Google Distance [29].

**Pointwise mutual information:** This approach is suggested by Turney [27] and depends on collecting data over very large corpora to find words' co-occurrences, then find the degree of statistical dependence using the following equation:

$$PMI - IR(w_1, w_2) = \log_2 \left( \frac{p(w_1 \text{ AND } w_2)}{p(w_1)p(w_2)} \right) \quad (2.1)$$

The probability  $p(w_i \text{ AND } w_j)$  can be found using NEAR query, which takes ten words occurrence window as follows:

$$p(w_i w_j) = \frac{hits(w_i NEAR w_j)}{WebSize} \quad (2.2)$$

Whereas the probability  $p(w_i)$  can be approximated as follows:

$$p(w_i) = \frac{hits(w_i)}{WebSize} \quad (2.3)$$

So equation 2.1 can be rewritten using 2.2 and 2.3 as follows:

$$PMI - IR(w_1, w_2) = \log_2 \left( \frac{hits(w_i) \times WebSize}{hits(w_i) hits(w_j)} \right) \quad (2.4)$$

$hits(w)$  denotes number of documents search engine returns when searching for  $w$ , and  $WebSize$  denotes number of words indexed by search engine, as defined by Chklovski [30]. The pointwise mutual information can be used to evaluate synonym candidates.

To measure the similarity between two text segments  $T_1$  and  $T_2$ , for each word in  $T_1$  and  $T_2$  the most similar word in other segment is determined using 2.1 equation. Then the computed similarities scores are combined using the following equation:

$$sim(T_1, T_2) = \frac{1}{2} \left( \frac{\sum_{w \in T_1} maxSim(w, T_2) \times idf(w)}{\sum_{w \in T_1} idf(w)} + \frac{\sum_{w \in T_2} maxSim(w, T_1) \times idf(w)}{\sum_{w \in T_2} idf(w)} \right) \quad (2.5)$$

Where  $idf(w)$  is the inverse document frequency of the word  $w$ .

**Latent Semantic Analysis:** Latent Semantic Analysis (LSA) was introduced by Landauer [28]. LSA extracts relations and deducts those of expected contextual usage of words in paragraphs and documents, depending on the assumption that semantically similar words occur in similar text pieces [25]. LSA has two main steps: building a matrix  $C$  that represents text, such that each row represents a unique word and each column represents the corresponding text. Then applying Singular Value Decomposition (SVD) on it to reduce number of columns while the rows similarity structure is kept as it is.

SVD decomposes the matrix Term  $C$  into three matrices as follow:

$$C = U \sum_k V^T \quad (2.6)$$

Where  $U$  and  $T$  are column orthogonal matrices and  $\Sigma$  is a diagonal  $k \times k$  matrix.

After finding the relation in the vector space, similarity between any two columns can be computed using the cosine similarity.

It worths to mention that LSA gets rid of some of the standard vector representation problems, such as sparseness and dimensionality, and makes it is possible to compute similarity between two documents without the need of external language semantic network or source.

The main drawback of using LSA to find similarity between document is that LSA models computed concepts can't be mapped to human readable concepts, which makes the models are not easily interpreted.

**Google Distance:** Cilibrasi and Vitanyi [29] proposed Google Distance as a measure of semantic similarity that depends on number of pages returned by Google search engine for words. The Normalized Google Distance (NGD) is defined as follows:

$$NGD(x, y) = \frac{\max(\log f(x), \log f(y)) - \log f(x, y)}{\log N - \min(\log f(x), \log f(y))} \quad (2.7)$$

Where  $f(x)$  equals to number of pages that contains  $x$ , and  $f(x, y)$  equals to number of pages that contains both  $x$  and  $y$  returned by Google search engine.

Based on the equation above, if  $x$  and  $y$  always occur together then their NGD equals to zero, and if never occur together then their NGD equals to infinity.

To measure the similarity between two keywords, use Google search engine to look for them apart, then use it again to look for them together and use equation 2.7. One can note that NGD ignores both word number and position of occurrence in the retrieved pages. Also NGD doesn't take Page Rank into consideration.

One of the major limitations in using corpus based similarity approaches is that it can't be used to find similarity between documents in different languages.

## 2.2 Knowledge Base-based Similarity

Knowledge base-based similarity depends on information drawn from semantic network, like WordNet, to measure similarity between two concepts [26]. Budanitsky and Hirst [31] discuss the following Knowledge-based measures:

**HirstSt-Onge:** According to Hirst and St-Onge [32] the strength of the relationship between two WordNet concepts depends on the length and direction of the path between their synsets. Based on this assumption three types of relationship are defined: Extra Strong, Strong and Medium relationships. Extra strong relationship "*occurs between a word and its literal repetitions*". Strong relationship occurs in three types. The first type is when the two concepts have a common synset. When there is a relation, such as antonymy or similarity, between the synsets of each concept then this is a Strong relationship of type two. The third

type occurs when there is a link between a synset of each word if a word is a phrase that contains the other.

When an allowable path connects a synset of each word, then we have a medium strong relationship. Medium strong relation detection is more complex than the other relations, as one should identify the allowable path in order to decide whether a relation is a medium strong one or not. Weight is defined also for this type of relation according to the following equation:

$$weight = C - \text{length} - K * \text{number of changes of direction} \quad (2.8)$$

Where C and K are constants.

**LeacockChodorow:** Leacock and Chodorow [33] approach depends on the shortest path between two concepts (len) to compute similarity. Shortest path can be found by counting nodes. The maximum depth of the taxonomy (D) is used then to scale the similarity as follows:

$$sim_{LC}(c1, c2) = -\log\left(\frac{len(c1, c2)}{2D}\right) \quad (2.9)$$

**Wu and Palmer:** Wu and Palmer [34] proposed the following similarity measure:

$$sim_{WP} = \frac{2 \times depth(LeastCommonSubsumer(LCS))}{depth(concept1) + depth(concept2)} \quad (2.10)$$

Where LCS is the least common subsumer. The Information Content of the LCS of two concepts can be measured using **Resnik** approach [35] as follows:

$$IC(C) = -\log P(C) \quad (2.11)$$

Resnik also proposed the following similarity metric:

$$sim_R = IC(LCS) \quad (2.12)$$

The probability  $P(C)$  is the probability of encountering an instance of concept C in the taxonomy. This means that the result similarity value has a minimum of zero. But it doesn't converge to a maximum value, as it's maximum value depends on the semantic network size.

**Lin Semantic Similarity in Taxonomy:** Lin [36] scaled the information content by the sum of the information contents of the two concepts, and proposed the following equation to measure similarity between two concepts  $C_0, C_1$

$$sim_{Lin} = \frac{2 \times \log P(C_0)}{\log P(C_0) + \log P(C_1)} \quad (2.13)$$

Using WordNet or other Knowledge bases to represent texts is limited to individual words, and can't be easily expanded to find similarity between long texts. This also removes the context role in word sense disambiguation. The last thing about using such approaches is that the WordNet representation of text is not weighted, which means it completely ignores the words frequencies [3].

## 2.3 Lexico-Syntactic Patterns-based Similarity

Panchenko, Morozova and Naets [37] proposed a semantic measure that is based on both lexico-syntactic patterns and Corpus. This approach uses 18 Finite State patterns, that are based on linguistic knowledge. These patterns are [37]

1. such NP as NP, NP[,] and/or NP
2. NP such as NP, NP[,] and/or NP
3. NP, NP [,] or other NP
4. NP, NP [,] and other NP
5. NP, including NP, NP [,] and/or NP
6. NP, especially NP, NP [,] and/or NP
7. NP: NP, [NP,] and/or NP
8. NP is DET ADJ.Superl NP;
9. NP, e. g., NP, NP[,] and/or NP;
10. NP, for example, NP, NP[,] and/or NP;
11. NP, i. e.[,] NP;
12. NP (or NP);
13. NP means the same as NP;
14. NP, in other words[, ] NP;
15. NP, also known as NP;
16. , NP, also called NP;
17. NP alias NP;
18. NP aka NP.

The first step in this approach is applying these patterns on a corpus. The output of this step is concordances of the form:

- *such non-caffeinated [drinks] as [coconut water][PATTERN=1]*
- *object oriented[language], such as [Java], [C#], and [Python][PATTERN=2]*

The nouns in the the square bracket are lemmatized with dictionary help in the second step. In the third step, each concordance with at least two terms from input vocabulary C is selected. The similarity matrix S is constructed then, where each entry  $s_{i,j}$  is equal to the frequency of terms in the square brackets within the same concordance. In the last two steps, the word pairs are ranked and the similarity scores are normalized.

Word pairs are ranked using some metrics, depends on the frequencies of words in different contexts.

The main challenges in using such approaches are huge effort and resources to cover the different patterns of the language lexicon.

## 2.4 Explicit Semantic Analysis

Gabrilovich and Markovitch [9] introduced the Explicit Semantic Analysis (ESA) to compute the semantic relatedness between two texts. The idea behind ESA is to represent the text as a weighted vector of natural concepts. Wikipedia is a good source to get these natural concepts. Explicit Semantic Analysis sometimes is classified as corpus-based approach, but we prefer to present it in a separate section in details because it has a main role in our approach.

The first step to map each fragment of the free text to its weighted ordered concepts. Semantic Interpreter is used for that where its output is an interpretation vector of concepts. As the interpretation vectors are constructed, computing similarity can be done using Cosine Similarity. For each text fragment, the related concepts are ranked according to their relevance using conventional text classification algorithms. Each of these concepts is represented as attribute vector of words that occur in the corresponding articles. The word-concept strength is reflected in the Term Frequency - Inverse Document Frequency (TF-IDF) weight associated in each word, like the “word” vector below.

$$\text{“word”} : [< \text{concept0}, \text{weight0} >, < \text{concept1}, \text{weight1} >, < \text{concept2}, \text{weight2} >, \dots] \quad (2.14)$$

A trivial example might be:

$$\text{“Mars”} : [< \text{planet}, 0.90 >, < \text{Solarsystem}, 0.85 >, < \text{jupiter}, 0.30 >, \dots] \quad (2.15)$$

Documents are represented as a combination of individual word vectors derived from the words within a document. Figure 2.1 shows the flow of ESA Semantic Interpreter.



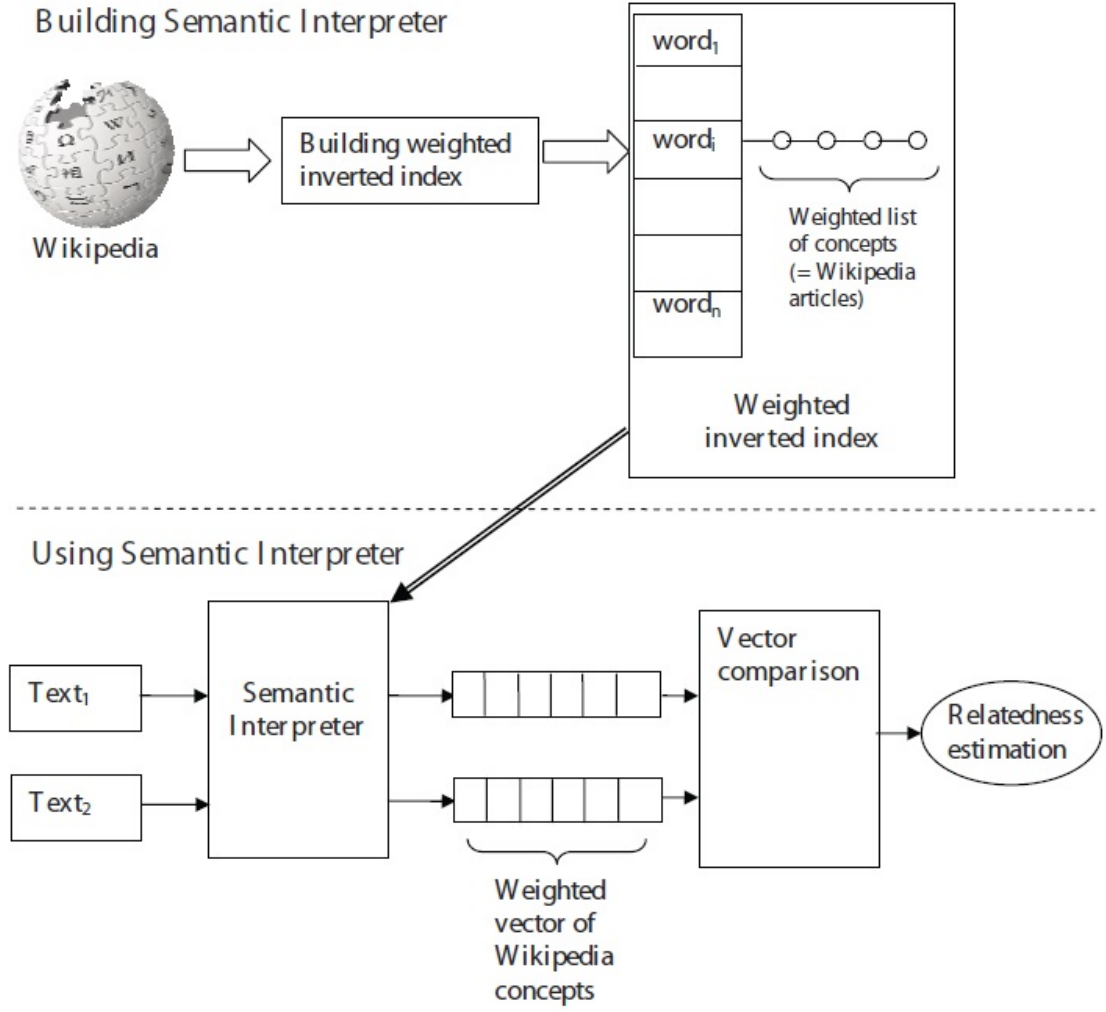


Figure 2.1: ESA Semantic Interpreter[9]

For better performance, an inverted index is constructed to perform word-concept mapping. Concepts with low weights for a given word are dropped out.

Gotttron [38] express the ESA similarity computation in equations. Take  $D$  as a collection index of  $V$  vocabularies with  $m$  terms occur. A document  $x$  is represented as a concept vector  $u$  such that

$$u^T = (< d_1, x >, < d_2, x >, \dots, < d_n, x >) \quad (2.16)$$

The inner product  $< d_i, x >$  reflects the similarity between  $x$  and  $d_i$ , i.e the association between  $x$  and the concept defined by  $d_i$ . Based on the previous definitions

the similarity between two documents  $x$  and  $y$ , with concepts vectors  $u$  and  $v$ , can be computed using the cosine similarity as follows:

$$sim_{ESA}(x, y) = cos(u, v) = \frac{< u, v >}{|u|. |v|} \quad (2.17)$$

$$sim_{ESA}(x, y) = \frac{1}{|u|. |v|} \sum_{i=1}^n < d_i, x > . < d_i, y > \quad (2.18)$$

Cosine similarity plays an important role in using ESA in CLIR. Sorg [39] makes use of an Wikipedia important feature that articles are linked across languages to adapt ESA in CLIR. Such feature means there are cross-language links that map an article to it's correspondence in another language. Sorg's approach depends on two mapping functions,  $\phi$  and  $\Psi$ .  $\phi$  map input text  $T$  to it's interpretation vectors.

$$\phi_k : T \rightarrow \Re^{W_k} \quad (2.19)$$

$$\phi_k(t) := < v_1, \dots, v_{W_k} > \quad (2.20)$$

Where  $W_k$  is Wikipedia database for language  $L_k$ , and the  $\phi$  vector components represent the association strength between the text  $T$  and the corresponding Wikipedia article.

$\Psi$  indexes a text in a language with respect to any other target language by transforming it's vector  $\phi(text)$  to a corresponding vector, which is spanned by article in the target language.

$$\Psi_{i \rightarrow j} : \Re^{W_i} \rightarrow \Re^{W_j} \quad (2.21)$$

To find the  $j$ -language ESA representation of text in language  $i$ , it's needed to compute  $\Psi_{i \rightarrow j}(\phi_i(text))$ .

Table 2.1 below shows the top ranked Wikipedia articles for the query "Scary Movies" in English and German ("Horrorfilme").

Table 2.1: Top ranked Wikipedia Articles in English and German (mapped to English) for query “Scary Movies”

Number	English Article	German Article
1	Scary Movie	Horror
2	Horror	Audition (disambiguation)
3	Scary Movie 3	Dark Water
4	Kazuo Umezu	Candyman
5	James L. Venable	Splatter film
6	Horror and terror	Prophecy (film)
7	Regina Hall	Wolfen (film)
8	Little Shop of Horrors	The Borrower
9	The Amityville Horror (1979 film)	Brotherhood of Blood
10	Dimension Films	Lionel Atwill

The actual overlapping is illustrated in table 2.2 below, which shows the positions of some of the Wikipedia articles in the English and German vectors for the same query “Scary Movies”.

Table 2.2: The positions of some Wikipedia articles in the ranked vectors in English and German (mapped to English) for query “Scary Movies”

Article	Position in Ranked English Vector	Position in Ranked German Vector
Scary Movie	1	555
Horror	2	1
Scary Movie 3	3	288
Scary Movie 2	4	619
The Amityville Horror (1979 film)	10	262
Scary Movie 4	12	332
Horror film	15	15
Horrorpunk	16	353
Jon Abrahams	23	235
Poltergeist (film series)	29	542

One can see that the both vectors have common articles, although the positions

of these articles are different. To find the similarity between query in language  $i$  and document in language  $j$ , cosine similarity can be used for the constructed query and document vectors as follows:

$$\cos(q_i, d_j) = \cos(\phi_i(q_i), \Psi_{j \rightarrow i}(\phi_j(d_j))) \quad (2.22)$$

ESA depends on Wikipedia to represent documents, which needs continuous updates due to the number of articles added to Wikipedia or deleted from it. One more challenge is that for cross-lingual document similarity, articles that exist in one language only can't be used to represent documents.

# Chapter 3

## Proposed Approach

### 3.1 Cross Lingual Document Similarity

To find the similarity between two documents in different languages, CL-ESA based approach will be used. This approach will basically be done in three steps:

- Represent each concept in the two documents with a set of Wikipedia categories, YAGO classes, DBpedia ontology classes. Here Wikipedia categories might have the least potential because:
  1. YAGO has a better taxonomy as WordNet was used to resolve conflicts in classes extracted from Wikipedia ontology [40]
  2. DBpedia class hierarchy is well structured and doesn't rely on community effort
- After representing all the documents concepts, multilingual mapping of categories or classes will be required (to get unified vectors)
- The last step is to compute the ESA vectors similarity

#### Prepare Wikipedia Data

Wikipedia data and metadata need to be extracted in order to represent documents as Wikipedia concepts. Data, such as Wikipedia page text and introduction, are required to create the inverted index which will be used to represent documents as Wikipedia concepts. Metadata is used generally to filter out concepts or to find relations between them. We started with declaring the required Wikipedia data and metadata fields, then we scanned the available sources to retrieve those fields from.

The required fields for Wikipedia Page are :

- Title
- Wikibase-Item: is used to uniquely identify the page
- Text: is used to create the inverted index
- Language
- Categories: are used to create Category vector
- Author: is used for future analysis
- User: is used for future analysis
- Namespace: is used to check whether this Page is an Article or not. Articles belong only to the main namespace (0)
- Disambiguation Page: is used to check whether this page is a disambiguation page or not. Disambiguation pages are used in Wikipedia to resolve the conflicts that occur when articles about two or more different topics could have the same natural title [41]
- Redirect Page: is used to check whether this page is a redirect page or not. Redirect pages automatically send the visitors to another Wikipedia page [42]
- Introduction: is used to get the article introduction for similarity tests
- InfoBox: is used for future analysis

And the required fields for Category are:

- Title
- Wikibase-Item: is used to uniquely identify the Category
- Language
- Categories: are used to for creating Category Links
- Redirect Page

To extract those fields, all Wikipedia pages for article and categories in Arabic and English need to be imported using either Wikipedia Dumps or Wikipedia API.

**Wikipedia Dumps** Wikimedeia<sup>1</sup>, which is an American non-profit organization that hosts Wikipedia, provides downloadable dumps for Wikipedia that are generated periodically<sup>2</sup>. These dumps contain data and metadata for Wikipedia, Wikidata and other related projects, and available in different languages. These dumps are also available in different formats, like static HTML dumps, gzipped SQL, JSON or XML, bzipped XML, and 7zipped XML [43].

The downloads were scanned to find the candidate ones that may contain the required data and metadata in categories and pages. The following candidate dumps were found:

- Category information (SQL): contains category id, category title, category pages, category subcategory count, and category files
- Wiki category membership link records (SQL): contains links between categories and articles
- List of page titles in main namespace (Text): contains the titles of all pages in the main namespace (articles)
- Articles, templates, media/file descriptions, and primary meta-pages, in multiple bz2 streams, 100 pages per stream (XML): contains most of the pages metadata, however some required items are missing such as introduction, last revision user, is redirect page and Wikibase-Item
- First-pass for page XML data dumps (xml): contain no page text
- Extracted page abstracts for Yahoo (xml): contains only page abstracts

We found that the required articles and categories data and metadata can't be extracted from those dumps, so we decided to create our own dumps using Wikipedia API.

**Wikipedia API** MediaWiki<sup>3</sup> is a free and open source software wiki package written in PHP and provides an action web service application programming interface (API) that allows developers to access some wiki-features like authentication, page operations, and search. Wikipedia API<sup>4</sup> is part of the MediaWiki API package, and can be used to access Wikipedia data.

To get Wikipedia pages in Arabic and English, all page titles in each language need to be retrieved using *query* action with *allpages* parameter, then another API

---

<sup>1</sup><https://www.wikimedia.org/>

<sup>2</sup>Dumps can be found in <https://dumps.wikimedia.org/enwiki/20190320/>

<sup>3</sup><https://www.mediawiki.org/wiki/MediaWiki>

<sup>4</sup><https://en.wikipedia.org/w/api.php>

request is invoked to retrieve page content by title. Since the maximum number of retrieved page titles in each request is 500, one needs to invoke the API request several times to retrieve all titles. Request 3.1 is used to retrieve the first 500 page titles in English Wikipedia.

```
https://en.wikipedia.org/w/api.php?action=query&format=xml&list=allpages&
apnamespace=0&apfilterredir=nonredirects&aplimit=500
```

(3.1)

This request contains the following options and values:

- **action=query** type of request you need to invoke. Query is used to fetch data from and about MediaWiki
- **format=xml** the response format. XML is used here
- **list=allpages** the parameters for the query request. *allpages* is used to enumerate all pages sequentially in a given namespace
- **apnamespace=0** Wikipedia name space to retrieve data from. Articles can be retrieved from the main namespace (0) whereas categories can be retrieved from namespace 14
- **apfilterredir=nonredirects** the titles to exclude. Here the redirect pages are filtered out
- **aplimit=500** number of titles to retrieve per request. 500 is the maximum

The response for this request would look like the one in 3.1 below.

```
<?xml version="1.0"?>
<api batchcomplete="">
<continue apcontinue="&quot;Central_New_York_Regional_Market&quot;
    ↪ ;"
continue="-||" />
<query>
<allpages>
<p pageid="3632887" ns="0" title="!!" />
<p pageid="600744" ns="0" title="!!!" />
<p pageid="2556962" ns="0" title="!!! (album)" />
<p pageid="55029148" ns="0" title="!!! (disambiguation)" />
<p pageid="2978668" ns="0" title="!Action Pact!" />
<p pageid="47197315" ns="0" title="!Arriba! La Pachanga" />
```



```

<p pageid="1921683" ns="0" title="!Hero" />
<p pageid="6893310" ns="0" title="!Hero (album)" />
<p pageid="58164345" ns="0" title="!Kora Wars" />
<p pageid="2516600" ns="0" title="!Kung languages" />
<p pageid="22602473" ns="0" title="!Oka Tokat" />
.
.
<p pageid="16250549" ns="0" title="&quot;C&quot; Is for Corpse"
  ↪ />
</allpages>
</query>
</api>

```

In this response, the *continue* tag contains two attributes; *continue* which tells whether there is still data to retrieve or not and *apcontinue* which contains the first title in the next title list to retrieve. The request below then is kept invoked till the value of the *continue* attribute in the response is null.

```

https://en.wikipedia.org/w/api.php?action=query&format=xml&list=allpages&
apnamespace=0&apfilterredir=nonredirects&aplimit=500&continue=-||&apcontinue=
Central_New_York_Regional_Market

```

(3.2)

One can see that two new parameters are appended in this request compared to the first one.

- **continue=-||** indicates that you need to continue retrieving data from a previous request
- **apcontinue=Central\_New\_York\_Regional\_Market** contains the value of the first title in the title list to retrieve. This value is the same of the *apcontinue* attribute value in the last response.

Each response is parsed using XML XPATH parser and titles are stored. After retrieving pages titles for articles and categories in Arabic and English, the Category page data and metadata for each Category title is retrieved using the following request:

```

https://LANGUAGE.wikipedia.org/w/api.php?action=query&format=xml&titles=
CATEGORY_TITLE&redirects&prop=pageprops|categories&cclimit=500

```

(3.3)

Where:

- **LANGUAGE** the category language, and can be en or ar
- **action=query** type of request to you need to invoke. Query is used to fetch data from and about MediaWiki
- **format=xml** the response format. XML is used here
- **titles=CATEGORY\_TITLE** CATEGORY\_TITLE is the title of the category after appending “category:” or “:تصنيف” prefixes for English and Arabic categories, respectively
- **redirects** to return all redirect pages to that category
- **prop=pageprops** to get various page properties defined in the page content
- **categories** to list all categories the pages belong to
- **cllimit=500** to set maximum number of parent categories to list

The XML response 3.1 below is retrieved for category Asia from English Wikipedia.

```
<?xml version="1.0"?>
<api batchcomplete="">
<query>
<normalized>
<n from="category:asia" to="Category:Asia" />
</normalized>
<pages>
<page _idx="697006" pageid="697006" ns="14" title="Category:Asia"
  ↪ >
<pageprops wikibase_item="Q5610083" />
<categories>
<cl ns="14" title="Category:Afro-Eurasia" />
<cl ns="14" title="Category:Commons category link is on Wikidata"
  ↪ />
<cl ns="14" title="Category:Continents" />
<cl ns="14" title="Category:Eurasia" />
<cl ns="14" title="Category:Wikipedia categories named after
  ↪ continents" />
</categories>
</page>
</pages>
</query>
</api>
```

After that, the article page for each title retrieved before is dumped using the same API . Request 3.4 below is used to retrieve page data and metadata

```
https://LANGUAGE.wikipedia.org/w/api.php?action=query&format=xml&titles=
PAGE_TITLE&redirects&prop=pageprops|categories|extracts|revisions&exintro=
&exsectionformat=plain&rvprop=size|user|content&cclimit=500"
```

(3.4)

Where:

- **LANGUAGE** the category language, and can be en or ar
- **action=query** type of request to you need to invoke. Query is used to fetch data from and about MediaWiki
- **format=xml** the response format. XML is used here
- **titles=PAGE\_TITLE CATEGORY\_TITLE** is the title of the page
- **redirects** to return all redirect pages to that page
- **prop=pageprops** to get various page properties defined in the page content
- **categories** to list all categories the pages belong to
- **extracts** to return plain-text or limited HTML extracts of the given pages
- **revisions** to get revision information
- **exintro** to get the introduction section in a separate tag
- **exsectionformat=plain** to specify the format to retrieve the introduction in
- **rvprop=size—user—content** to specify the page revision properties to retrieve
- **cclimit=500** to set maximum number of parent categories to list

The XML response 3.1 below is retrieved for page Birzeit University.

```

<?xml version="1.0"?>
<api batchcomplete="">
<query>
<normalized>
<n from="Birzeit_University" to="Birzeit University" />
</normalized>
<pages>
<page _idx="2247791" pageid="2247791" ns="0" title="Birzeit
    ↪ University">
<pageprops page_image="Birzeit_University_seal.svg" wikibase_item
    ↪ ="Q510029" />
<categories>
<cl ns="14" title="Category:1924 establishments in Mandatory
    ↪ Palestine" />
<cl ns="14" title="Category:All accuracy disputes" />
<cl ns="14" title="Category:All articles with unsourced
    ↪ statements" />
<cl ns="14" title="Category:All articles with vague or ambiguous
    ↪ time" />
<cl ns="14" title="Category:Articles containing Arabic-language
    ↪ text" />
<cl ns="14" title="Category:Articles with Arabic-language
    ↪ external links" />
<cl ns="14" title="Category:Articles with disputed statements
    ↪ from May 2015" />
<cl ns="14" title="Category:Articles with unsourced statements
    ↪ from September 2015" />
<cl ns="14" title="Category:Birzeit" />
<cl ns="14" title="Category:Birzeit University" />
<cl ns="14" title="Category:Commons category link from Wikidata"
    ↪ />
<cl ns="14" title="Category:Coordinates on Wikidata" />
<cl ns="14" title="Category:Educational institutions established
    ↪ in 1924" />
<cl ns="14" title="Category:Instances of Infobox university using
    ↪ image size" />
<cl ns="14" title="Category:Pages using deprecated image syntax"
    ↪ />
<cl ns="14" title="Category:Universities and colleges in the
    ↪ State of Palestine" />

```

```

<cl ns="14" title="Category:Vague or ambiguous time from
↳ September 2015" />
<cl ns="14" title="Category:Wikipedia articles with GND
↳ identifiers" />
<cl ns="14" title="Category:Wikipedia articles with ISNI
↳ identifiers" />
<cl ns="14" title="Category:Wikipedia articles with VIAF
↳ identifiers" />
<cl ns="14" title="Category:Wikipedia articles with WorldCat-VIAF
↳ identifiers" />
</categories>
<extract xml:space="preserve">&lt;p&gt;&lt;b&gt;Birzeit
↳ University&lt;/b&gt; (Arabic: &lt;span lang="ar" dir="rtl"&
↳ gt;&lt;big&gt; &lt;/big&gt;&lt;/span&gt;), often
↳ abbreviated as &lt;b&gt;BZU&lt;/b&gt;, is a public
↳ university located in Birzeit, West Bank, near Ramallah.
↳ Established in 1924 as an Elementary School for girls,
↳ Birzeit became a University in 1975.&lt;/p&gt;&lt;p&gt;
↳ Birzeit University, with the highest admission averages
↳ among other Palestinian universities, offers graduate and
↳ undergraduate programs in information technology,
↳ engineering, sciences, social policy, arts, law, nursing,
↳ pharmacy, health sciences, economics, and management. It
↳ has 9 faculties, including a graduate faculty. These offer
↳ 47 B.A. programs for undergraduate students and 26 M.A.
↳ programs for graduate students. As of 2018, around 14,000
↳ students are enrolled in the university's bachelor's,
↳ master's and PhD programs.&lt;/p&gt;</extract>
<revisions>
<rev user="LilHelpa" size="11761" contentformat="text/x-wiki"
↳ contentmodel="wikitext" xml:space="preserve">{{Infobox
↳ university
|name = Birzeit University
|native_name =
|native_name_lang = ar
|image = Birzeit University seal.svg
|image_size = 150px
|established = 1975
|type = Public
|image_name =

```

```

|caption =
|president = Abu al ameer (or Abdul Latif Abu Hijleh)&lt;ref&gt;[
    ↪ http://www.birzeit.edu/en/about/president-office Presidents
    ↪ Office], Birzeit University&lt;/ref&gt;
|campus = Urban&lt;br /&gt;800 dunums (200 acres)
|undergrad = 8,465
|postgrad = 1,388
|academic_staff = 617
|motto = Building a Better Palestinian Future
|city = [[Birzeit]]
|country = [[West Bank]]
|website = [http://www.birzeit.edu www.birzeit.edu]
|affiliations = [[Mediterranean Universities Union|UNIMED]], [[
    ↪ Association of Arab Universities|AARU]]
|logo = [[File:Birzeit University logo.svg|250px]]
}}
.
.
.
</rev>
</revisions>
</page>
</pages>
</query>
</api>

```

The response for each category and page is stored as a XML text file. Table below shows total number of files for each language.

Table 3.1: Total Number of Retrieved Wikipedia Pages and Categories in Arabic and English

Language/Concept	Categories	Pages
Arabic	171443	834412
English	1370816	11306222

Data retrieved in June, 2017

## Parse Wikipedia Data Dumps

After preparing Wikipedia dumps, we need to start parsing then storing them in database. Since each Page and Category is stored in XML file in Wikipedia

format, XML parsers are needed to extract the required data from those file. We built XML parsers based on XPath in Java that loops over all XML files in the dump directory and extract the tags and attributes values based on predefined XPath expressions.

To parse pages, three parsers are developed: Metadata Parser, Data Parser and Info Box Parser and is done in the following steps:

1. Parse page metadata using Page Metadata Parser. Page title, Wikibase-Item, namespace, disambiguation page, category titles, revision User and redirect page are all extracted directly
2. Check if this page is a complete article. A complete article should have the following properties:
  - Wikibase-Item is not null or empty string
  - Namespace is 0
  - Not a disambiguation page
  - Not a redirect page

If the page is not a complete article, stop and skip it.

3. Parse Page Data using Page Data Parser. Page content and introduction content are extracted directly in this step, then each of them is parsed to extract the remaining fields.

The page content is cleaned initially by doing the following data cleaning steps:

- Remove all non-4 bytes characters. These mostly are either non-printable characters or symbols
- Remove references tags. These tags starts with “<ref”
- Remove square brackets

Then each line in the washed content is parsed as follows:

- Skip line if it starts with “File:” or “:ملف”. Such lines contain technical descriptions for the multimedia elements
  - Skip line if it starts with |. Such lines contain extra HTML formatting for tables, and need special handling.
  - If line starts with “{{” and contains “Infobox” or “ص.م”, parse it using the InfoBox Parser and add it to the line pool. The InfoBox Parser simply extracts the texts from InfoBox parenthesis
  - If line doesn’t start with “{{”, doesn’t end with “}}”, doesn’t contain “=” or “references” or “مراجع”, and it is not the last line, then parse it using the InfoBox Parser and add it to the line pool
  - The line pool then is normalize using Apache Lucene and normalize content is stored. The article word count is calculated here by finding number of words in the normalized content
4. Create the inverted index for the normalized page content. This inverted index contains the tokens along with their frequencies

At this point, the page is completely parsed, and if it is a complete article, then the article object is ready to be inserted into database.

To parse category files, another simple XML parser is developed. The parser extracts title, Wikibase-Item, redirect page and parent categories then checks if this category is a complete category or not. Similar to complete page, a complete category should have the following properties:

- Wikibase-Item is not null or empty string
- Not a redirect category page

If the category is a complete category, insert it into database, else stop and skip it.



## Store Wikipedia Data in Database

Complete articles and categories need to be inserted into database to make them accessible for vector constructors. Due to the data volume and velocity, non-relational databases was the first option. And among all the NoSQL databases, Apache Cassandra<sup>5</sup>, which is a column oriented database, was the best option due to the following key features [44]:

- Scalablility
- Linear scale performance
- No single points of failure
- Read/write support across multiple data centers
- Cloud availability zones
- Tunable consistency
- Built-in data compression
- MapReduce support
- Primary/secondary indexes

Two Cassandra nodes were installed on Centos server to get a better reliability, and for each language, keyspace was create called wiki.en and wiki.ar with the following tables: article, category, info\_box, article\_category, category\_link, dictionary, posting, representaive\_category, representaive\_article and representative\_category\_link. Keyspace is created for each language to improve performance and maintenance on data. Appendix A shows the Cassandra Query Language (CQL) scripts used to create the wiki.en keyspace with tables.

To access Cassandra for read, write and summarization, Apache Spark<sup>6</sup> is used. Apache Spark is an open-source distributed and unified analytics engine for large-scale data processing[45]. Spark provides the ability to program the entire cluster with implicit data parallelism and fault tolerance [46].

For an application to use Spark, it needs to create a SparkContext object which coordinates independent sets of processes on a cluster. This SparkContext object needs be instantiated in the main program, which is called Spark Driver, and can connect to either Standalone, Apache MESOS, Hadoop YARN or Kubernetes cluster manager. These cluster managers are responsible of allocating resources for the

---

<sup>5</sup><http://cassandra.apache.org/>

<sup>6</sup><https://spark.apache.org/>

applications. When the SparkContext is connected, it acquires Spark Executors on the cluster nodes. Spark Executor is a process that runs computations and stores data. Then, SparkContext sends the application code, which is JAR file for example, to the executors, before sending them tasks to run. Figure 3.1 shows an overview for Spark Cluster. Here definitions of the main terms and components in

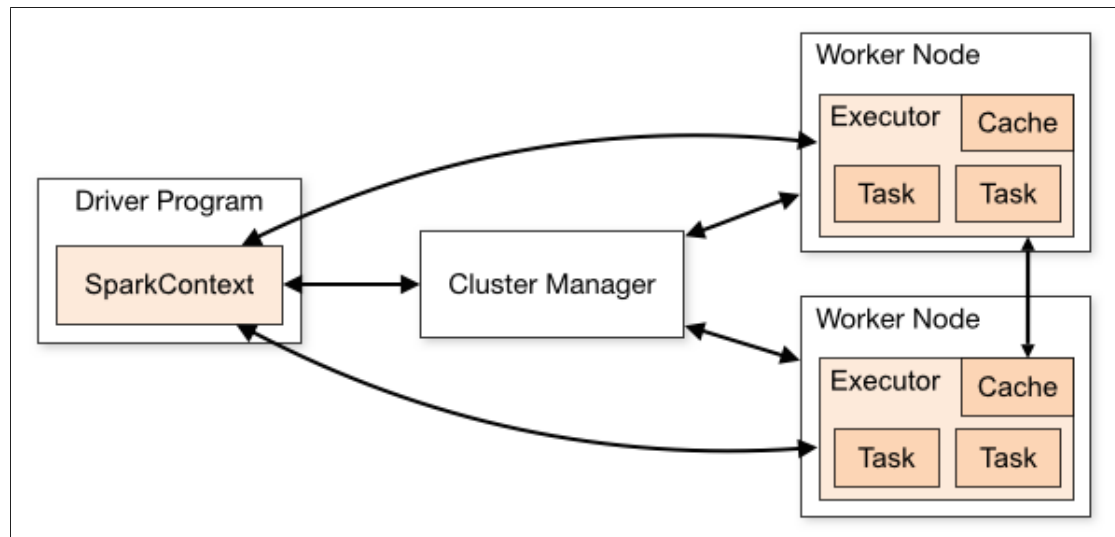


Figure 3.1: Spark Cluster Overview

Apache Spark:

- **Application** is a user program built on Spark
- **Application jar** is a jar containing the user's Spark application
- **Driver program** is the the process running the `main()` function of the application and creating the SparkContext
- **Cluster manager** is an external service for acquiring resources on the cluster
- **Deploy mode** is used to distinguish where the driver process runs. In “cluster” mode, the framework launches the driver inside of the cluster. In “client” mode, the submitter launches the driver outside of the cluster.
- **Worker node** is any node that can run application code in the cluster
- **Executor** is a process launched for an application on a worker node, that runs tasks and keeps data in memory or disk storage across them. Each application has its own executors

- **Task** is a unit of work that will be sent to one executor
- **Job** is a parallel computation consisting of multiple tasks that gets spawned in response to a Spark action (e.g. save, collect)
- **Stage** is a small set of tasks in the job. Job contains set of stages that depend on each other

To submit Spark application, Spark provides a *spark-submit* script, where you specify your application and configure the command line parameters passed to the driver and executor. The code snippet 3.1 is an example of the submit JAR file bash script.

```
spark-submit --class package.ClassName --master spark://hostname
  ↪ :7077 --deploy-mode cluster --conf "spark.driver.
  ↪ extraJavaOptions=-XX:+UseG1GC -Dwiki.property.file.path=
  ↪ wiki.properties" --conf "spark.executor.extraJavaOptions=-
  ↪ XX:+UseG1GC -Dwiki.property.file.path=wiki.properties" Wiki
  ↪ -0.0.1-SNAPSHOT-jar-with-dependencies.jar
```

After submitting the application, it can be monitored in Spark Web UI, which shows details about the running applications, workers, executors and stages along with the occupied resources [47].

For this work, Spark is installed on Centos server with Workers each with 2 executors and configured to run in Standalone Cluster Mode. This mode is used for its simplicity.

The different objects created for the parsed pages and categories are converted into Spark data frames which then inserted by the Spark Session created by the Spark Context object. Table 3.2 shows some of the statistics about the stored data for both Arabic and English.

Table 3.2: Statistics about the Stored Data

Statistics/Language	Arabic	English
Total # of Articles	433381	4348135
Total # of Categories (Linked with Articles)	146166	1224122
Total # of InfoBoxes	361377	6772021
Total # of Article Categories	4148892	31210309
Total # of Words in Dictionary	1860957	15214188
Average # of Unique Words per Article	178.11	282.56
Average # of Unique Words per Category	1914.99	2701.10
Average # of Categories per Article	9.57	7.18
Average # of Articles per Category	28.39	25.61

### Representative Articles and Categories

The first step in representing each word as a vector of Wikibase-Items, is to find a common set of Wikibase-Items in Arabic and English. This is needed for both Article and Category-based similarity.

**Representative Articles** Representative article set is selected by finding the common articles in Arabic and English. Each article has a Wikibase-Item, and the common articles is the result of intersect all the article Wikibase-Items in Arabic and English.

There are 433381 and 4348135 articles in the Arabic and English keyspaces, respectively. The results of the intersection is 239443 articles. This means we have 239443 articles in the representative article set.

**Representative Categories** Selecting the representative category set is more complicated than selecting the representative article set, due to two reasons:

- There is no direct relationship between words and categories. If one needs to find what are the categories of a specific word, he needs to find the articles that contain this word, then find the categories of these articles
- The category system in Wikipedia is not mature enough. An article may belong to category B, however it belongs only to Category A in Wikipedia, which is a parent or grand parent category for Category B.

The first step is to get the categories for both Arabic and English. The total number of categories in Arabic is 146166 and in English is 1224122. We started

with exploring Arabic categories and found that some of the categories have insufficient number of articles and other have too many articles, so we had to filter them by number of articles. Categories with 0, 1 or more than 1000 articles were excluded. The upper threshold 1000 was estimated based on the category-article distribution, and may need more specific experiments to find the best value to use. After removing those categories, number of Arabic categories is reduced to 81351.

The next step is to filter categories by name. We found that some categories are less informative than others, such as those talking about people born in a certain year. Other categories were created to group articles based on some Wikipedia internal properties, such to group those articles need improvement. To do that, we implemented the following filtration:

- Remove categories with numbers at the end of its names
- Remove Arabic categories with “تصنيف : بذرة” or “stub” in its names

To this point, we have worked on Arabic categories only, where some of them do not exist in the English keyspace. To have one common category set, common categories in both languages are extracted next by intersecting the Wikibase-Items of the Arabic categories with the Wikibase-Items of the English ones. Number of common categories is 110705. Then the Wikibase-Items of the common category set are intersected with the ones from the filtered category set. Number of common filtered categories is 47017.

Experimental results show that another stage of category filtration is needed. The informative of categories had to be revised. There are two approaches here:

- Remove words from that exist in a large number of categories. Such words might be considered as stop words, and removing them might improve the similarity results
- Remove categories that contain large numbers of unique words. Such categories might have less informativity than others

Words statistics in categories are retrieved, and those words exist in certain number of categories are filtered from the queries. Different thresholds for number of categories are used, but no significant improvement is observed.

To filter categories that contains large number of unique words, category-word statistics were retrieved for the common category set in Arabic and English. Table 3.3 shows the word count statistics for the initial representative category set in both Arabic and English.

Table 3.3: Representative Categories Word Count

Language	Minimum Word Count	Average Word Count	Maximum Word Count
Arabic	31	2178.1	48436
English	23	7602.6	282828

After retrieving number of words per category, categories with number of words greater than some thresholds were excluded incrementally from the representative category set and similarity tests were executed for each set. Table 3.4 shows number of remaining representative categories after excluding those categories with number of unique words greater than the ones specified for Arabic and English.

Table 3.4: Representative Categories

Maximum Word Count Per Category in Arabic	Maximum Word Count Per Category in English	Number of Representative Categories
-	-	47012
2028	7747	24825
1412	5867	19035
995	4412	13788
856	4045	12076
672	3269	9178

## Constructing Vectors

In order to perform similarity tests for queries and collections, TF-IDF vectors should be generated for each query and document in the collection. And to make a comparison between article and category based similarities, two TF-IDF vectors are created: article and category for each document.

The TF-IDF weighting assign each term (t) a weight based on its frequency in the document (d)  $tf(t, d)$  and in the collection  $idf_t$ , as follows:

$$TF - IDF = tf_{t,d} \times idf_t \quad (3.5)$$

Where  $tf_{t,f}$  is the term frequency, which represents the frequency of the term in the document. In this work, we used the weighted term frequency as tf, which is given by

$$wf_{t,d} = \begin{cases} 1 + \log tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.6)$$

and  $idf_t$  is the inverse document frequency. Document frequency,  $df_t$  represents the number of documents in the collection that contain a term  $t$ . And the inverse document frequency is given by

$$idf_t = \log \frac{N}{df_t} \quad (3.7)$$

Where  $N$  is number of the documents in the collection [48].

The first step in generating both article and category TF-IDF vectors for a document, is to create the inverted index for its tokens, which is done in the stages illustrated in Figure 3.2. The inverted index contains the normalized words and

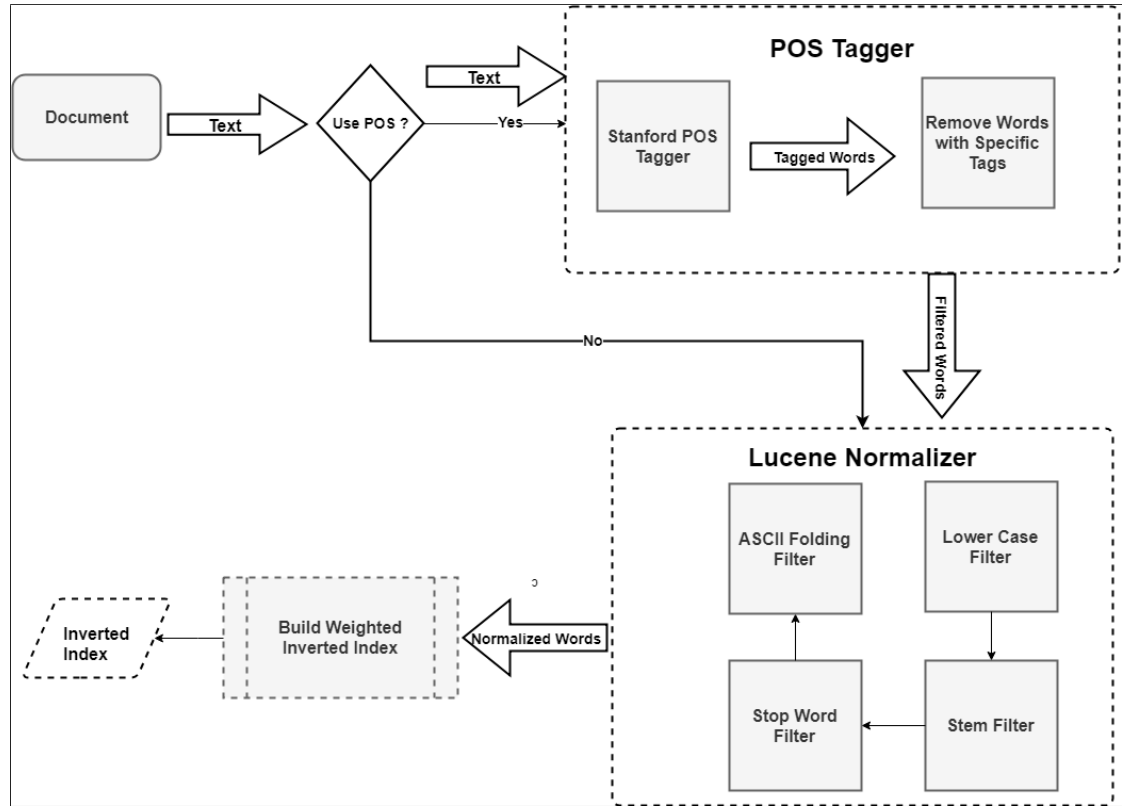


Figure 3.2: Building Weighted Inverted Index for Documents

its frequency in the document. In-house inverted index implementation is used, which has a method that takes all the document normalized words, then create a posting for each of them along with frequency. To normalize words, Apache Lucene<sup>7</sup> is used and includes:

<sup>7</sup><http://lucene.apache.org/>

- Porter Stem Filter for English and Arabic Stem Filter for Arabic. The Porter Stem Filter transforms the token stream as per the Porter stemming algorithm.
- ASCII Folding Filter to convert alphabetic, numeric, and symbolic Unicode characters which are not in the first 127 ASCII characters (the "Basic Latin" Unicode block) into their ASCII equivalents, if one exists.
- Lower Case Filter for English
- Stop Word Filter with Lucene default stop word lists in both Arabic and English. The default stop word list contains 162 words in Arabic and 33 words in English.

The input of the Lucene normalizer is the whole document text and it returns a list of normalized words. For Category TF-IDF vectors, POS filter is used in most of the tests. To apply this filter, Stanford Part-of-Speech Tagger (POS Tagger) [49] is used. POS Tagger is applied on both the query and the collection text before apply Lucene, then those words that are tagged as one of the POS types listed in table 3.5 are removed.



Table 3.5: POS Tags of the Terms Removed from Documents

<b>Tag</b>	<b>Description</b>
WDT	WH-determiner
WP	WH-pronoun
WP\$	WH-pronoun, possessive
WRB	Wh-adverb
UH	Interjection
SYM	Symbol
RP	Particle
TO	To
PDT	Pre-determiner
POS	Genitive marker
PRP	Pronoun, Personal
PRP\$	Pronoun, Possessive
MODAL	Modal Auxiliary
MD	Modal Auxiliary
LS	List Item Marker
IN	Preposition or Conjunction, Subordinating
EX	Existential There
DT	Determiner
CC	Conjunction, Coordinating
CD	Numeral, Cardinal

Tag descriptions are explained in Part-of-speech tagging guidelines for the Penn Treebank Project [50]

It is worth to mention here that for performance purposes, POS Tagger in category vectors is applied before word normalization. This is to avoid normalizing tokens that will be filtered by the tagger. The output of the POS Tagger is a list of words.

The output of the Lucene normalizer then is used to construct the inverted index.

To generate the article TF-IDF vector for the inverted index, all the Wikibase-Items of the representative articles that contain those words are retrieved from database with word frequencies in the articles. Then the weights for each word:Wikibase-Item pair are calculated as follows:

1.

$$wf_{t,f} = \begin{cases} 1 + \log(\text{word frequency in article}), & \text{if word frequency in article} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.8)$$

2.

$$idf_t = \log \frac{(\text{Representative article count})}{(\text{Number of unique representative articles that contain the word})} \quad (3.9)$$

3.

$$weight = (\text{word frequency in document}) \times wf_{t,f} \times idf_t \quad (3.10)$$

4. At this point, we have a map with the following component structure

$$word : Wikibase - Item : weight$$

so each distinct word in the document has a component with the structure above.

The last step is to group all the components by article Wikibase-Item and find the total weight for each group by adding its sub-weights. Words are now eliminated. The final result is a vector that has a set of unique Wikibase-Item weight components with the following structure:

$$Wikibase - Item : weight$$

And to generate the category tf.idf vector for the inverted index, all the Wikibase-Items of the representative categories that contain those words are retrieved from database with word frequencies in the categories. Since there is no direct link between words and categories in Wikipedia, we had to get words articles, then get article categories. The word frequency in category equals to the sum of all word frequency in articles which belong to this category. The weights for each word:Wikibase-Item pair are calculated as follows:

1.

$$wf_{t,f} = \begin{cases} 1 + \log(\text{word frequency in category}), & \text{if word frequency in category} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.11)$$

2.

$$idf_t = \log \frac{(\text{Representative category count})}{(\text{Number of unique representative categories that contain the word})} \quad (3.12)$$

3.

$$weight = (\text{word frequency in document}) \times wf_{t,f} \times idf_t \quad (3.13)$$

4. At this point, we have the following component:

$$word : articleWikibase - Item : weight$$

Similar to what we have in article tf.idf vector, the last step is to group all the components by category Wikibase-Item and find the total weight for each group by adding its sub-weights. Words are now eliminated as well. The final result is a vector that has a set of unique Wikibase-Item weight components with the following structure:

$$Wikibase - Item : weight$$

Apache Spark is used to generate the concepts vectors from Cassandra for all documents before each test by executing the steps above. Since documents are used in several tests and generating one vector may take up to 17 hours, each time a new vector is generated, the vector generator saves it to an XML text file. This is done by a component called Vector Generator. Next time, when this document is used in a similarity test, the test just needs to retrieve it from the XML file repository, instead of generating it again. This is done by a component called Vector Retrieval.

## Computing Similarity

To find the similarity between a query document and a set of documents in a collection, vectors need to be generated for the query and each of the documents. Then a proper similarity algorithm is used to find the similarity between the query vector and each document vector.

Cosine similarity is the standard way to quantifying the similarity between two vectors that represent two documents, and given by:

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| |\vec{V}(d_2)|} \quad (3.14)$$

Given that the numerator is the dot product of the vectors and the denominator is the product of the vectors length, the cosine similarity equation can be re-written as follows:

$$sim(d_1, d_2) = \frac{\sum_{i=1}^M V_i(d_1)V_i(d_2)}{\sqrt{\sum_{i=1}^M \vec{V}_i^2(d_1)}\sqrt{\sum_{i=1}^M \vec{V}_i^2(d_2)}} \quad (3.15)$$

Where M is the vector length[48].

## Results Evaluation

The input data for each similarity test is a set of queries in a language and a set of documents in the other language. The goal is to identify the document that is most related to the query; which could be a summary, a translation, or just a related document. The query result, which is usually the exact translation of the query or the most related document, is part of the collection documents.

In each test, the similarity between the category vectors of the query and each document in the collection is found, then the collection documents are sorted by the calculated similarity value. The rank of the query result is taken. This procedure is repeated for all queries.

To evaluate the performance of each test, number of queries which its result rank is  $\leq 10$ ,  $\leq 5$  and  $= 1$  is counted. These numbers represent the performance of the category or article set to represent the documents, which is reflected in finding the query results in the top retrieved documents.

Since finding the document most related to the query (by human judgment) as the first retrieved document (rank = 1) is more important than finding it in the first retrieved 5 or 10 documents (rank  $\leq 5$  and rank  $\leq 10$ , respectively). It might be unexpected to have a representative article or representative category set that gives the best results for an aggregate ranking in all tests, a new score has been used and given by:

$$\begin{aligned} \text{score} = & [(20\% \times \text{Number of documents with rank} \leq 10) \\ & + (30\% \times \text{Number of documents with rank} \leq 5) \\ & + (50\% \times \text{Number of documents with rank} = 1)] \\ & \div (\text{Total number of documents}) \end{aligned} \quad (3.16)$$

We set these weights as experimentally, changing them didn't make a big differences on the aggregate rank (will be showed later). The same tests are repeated for article vectors.

## 3.2 Access to Foreign Language Structured Data

Foreign language structured data are mainly stored as knowledge bases. This section proposes an approach to query knowledge bases in foreign languages. The approach can be used to generate Arabic structured and unstructured data from the results of the query to a knowledge base in a foreign language.

### 3.2.1 Query Foreign Language Structured Data

Knowledge bases store structured and unstructured data. This data can be either machine or human readable. YAGO, DBpedia and Wikidata are considered three of the most known knowledge bases. YAGO<sup>8</sup> is a knowledge base that is derived from Wikipedia, WordNet and GeoNames, with 10 million entities and 120 million facts about these entities [51]. The basic architecture of YAGO consists of entities and facts. YAGO extraction starts with extracting entities from a set of Wikipedia articles, extracts facts about these entities then create a taxonomy [40]. YAGO extraction is based on software modules called extractors and sets of facts stored in files called themes. Themes and other data are passed to extractors, which process them and yield output themes. Some other extractors postprocess the output of others [52]. Wikipedia articles become entities in YAGO, whereas Wikipedia Infoboxes are used to extract facts about these entities. Since YAGO3, YAGO became a multilingual knowledge base. The same English extractors were ported to each language. A dictionary was used to translate entities to their unique names and prefix them with their language code. Infoboxes attributes of different languages are mapped to YAGO schema. YAGO3 also proposed the principle of contribution to handle entities and facts that are unknown to English YAGO.

English YAGO taxonomy relies on Wikipedia categories. Category extractors were used to extract Wikipedia categories, filter out them then build the class hierarchy based on names. For foreign languages, Wikidata was used to extract the Categories keyword. The English category extractor was modified to extract foreign categories. Then a dictionary was used to translate the extracted categories to English. As overall integration, each entity belongs to at least one class, in which a WordNet taxonomy links these classes together. If a class couldn't be found for an entity, then it is abandoned. Foreign entities are either abandoned or mapped to one of the 102 English YAGO relations [40].

Birzeit University is an example of YAGO entities. It has 16 relations, table 3.6 shows some of them. YAGO provides four interfaces<sup>9</sup> to access data: graph

---

<sup>8</sup><http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago>

<sup>9</sup><https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/demo/>

browser, ontology browser, spotlx interface and SPARQL endpoint. Before using the SPARQL endpoint, one can explore the relations via Spotlx (shown in figure 3.3), which is an RDF-like interface, where data can be queried in the form of (subject, property, object). YAGO relations are not available in Arabic although mapping between multilingual relations to English is a mandatory step in YAGO3 extraction. As per email from F. Suchanek, YAGO Team(fabian@suchanek.name) in March 2016, the mapping between these multilingual facts and English is not available to the public. This means it is mandatory to translate the relations to English before querying YAGO.

**YAGO 2 spotlx**

**Query**

Id	Subject	Property	Object	Time	Location
? id0:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
? id1:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
? id2:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
? id3:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
? id4:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

query

Figure 3.3: YAGO Spotlx Interface

Table 3.6: Birzeit University Entity Relations in YAGO

Property	Value
linksTo	West Bank
linksTo	Ramallah
linksTo	First_Intifada
linksTo	Marwan_Barghouti
graduatedFrom	Yahya_Ayyash
hasLongitude	35.180705555555555
wasCreatedOnDate	1975
isLocatedIn	Birzeit Palestine
hasMotto	Building a Better Palestinian Future
skos:prefLabel	Birzeit University @eng
rdfs:label	Univiersite d' Birzeyt@wln
rdfs:label	Birzeit University@eng
rdfs:label	Universidad de Birzeit@spa
rdfs:label	@heb
rdfs:label	Universitt Bir Zait@deu
rdfs:label	Universit de Beir Zeit@fra
rdfs:label	”جامعة بيرزيت”@ara
rdfs:label	Universita di Bir Zeit@ita
hasWikipediaArticleLength	9479
hasWikipediaAnchorText	”السياسية”
hasWikipediaAnchorText	”بيرزيت”
hasWikipediaAnchorText	”فلسطين”
hasWikipediaAnchorText	”فتحي الشقاقي”
hasWikipediaAnchorText	”عزمي بشارة”
hasWikipediaAnchorText	”ستيفن هوكينج”
hasWikipediaAnchorText	”عمر العقاد”

Wikidata<sup>10</sup> is a free, open and multilingual knowledge base that acts as structured data central storage for Wikimedia project [53]. The idea behind creating Wikidata is to provide an up-to-date and machine readable data exports that can benefit search engines, social networks and many other web applications. Wikidata started reconciling the objects of articles in different languages. For this purpose, Wikidata made use of the crosslingual links connecting articles in different languages. For each article, a page is created in Wikidata to manage article pages in different languages called items [54]. Each article item in Wikidata has five sub-items: language, label, description and “also known as” (alias). Article page contains also Statements. Each statement contains a property with its value. Property-Value is the basic data model that Wikidata adopts to store structured data. For example, Birzeit university page in Wikidata<sup>11</sup> has 7 properties as in table 3.7.

Table 3.7: Birzeit University Item Properties in Wikidata

Property	Value
instance of	university
located in the administrative...	Birzeit
image	3157’31.0”N, 3510’50.9”E
topic’s main category	Category:Birzeit University
country	State of Palestine
inception	1923

The last section in Wikidata article page is the Identifiers, which are links for the article in other knowledge bases and data repositories. Table 3.8 shows the identifiers of Birzeit University.

Table 3.8: Birzeit University Item Identifiers in Wikidata

Identifier	Value
Freebase ID	m06z0pd
GND ID	6075351-1
VIAF ID	256740978
Twitter username	BirzeitUniv
grid global research id	grid.22532.34

Figure 3.4 shows a snapshot of Wikidata page of Birzeit University.

<sup>10</sup><https://www.wikidata.org/>

<sup>11</sup><https://www.wikidata.org/wiki/Q510029>



# Birzeit University (Q510029)

university

No aliases defined

▼ In more languages

Configure

Language	Label	Description	Also known as
English	Birzeit University	university	
Arabic	جامعة بيرزيت	No description defined	

More languages

## Statements

instance of

university

edit

▼ 0 references

+ add reference

Figure 3.4: Birzeit University Wikidata Page

Each Wikidata property has its own page, similar to item page. However, its page contains additional section for data type. Data type defines the domain of the property. For example, “postal code” is a string, “population” is a number and “head of government” is a Wikidata item. Wikidata contains 17,614,521 items and 2,195 properties [53].

In order to query Wikidata for data, two interfaces are available. The first one is for SPARQL<sup>12</sup> and provides a query graphical user interface to type SPARQL statements and execute them. The result is displayed as table and can be downloaded in different formats, such as JSON, CSV and TSV. The other interface is a web service.

To find number of properties that have either alias or description in Arabic, we developed an application that queries all the properties through Wikidata Web service. Results showed that only 46 out of 2,195 properties have Arabic alias, and 102 have Arabic description.

The relation between Wikidata and Wikipedia is still an open issue. Which one is the source of the other? are their facts linked? As per a discussion in Wikidata

<sup>12</sup><https://query.wikidata.org/>

Technical mailing list<sup>13</sup> on April 20, 2016 the data is not synced between Infoboxes and Wikidata. A lot of the data in Wikidata has been extracted from Infoboxes using users scripts and inserted into Wikidata, but not by Wikidata team. Also some Infoboxes make (partial) use of data from Wikidata, and some others are completely relying on Wikidata. The South Pole Telescope<sup>14</sup> probably is a good example of the articles with Infobox that is completely linked to Wikidata

DBpedia is a large-scale, multilingual knowledge base extracted from Wikipedia in 111 languages. DBpedia stores data of various topics and also points to external data sources with Resource Description Framework (RDF) links. Local organizations do the maintenance for each language chapter. DBpedia extraction framework, shown in figure 3.5, consists of four phases. In the first phase, Wikipedia pages are read from either Wikipedia dump or MediaWiki API. DBpedia then parses these pages and produces Abstract Syntax Trees. In the third phase, different purposes extractors are applied on the Abstract Syntax Trees to produce RDF statements. In the last phase, RDF statements are written to a sink [10]. DBpedia has different extractors to extract the different parts of Wikipedia pages.

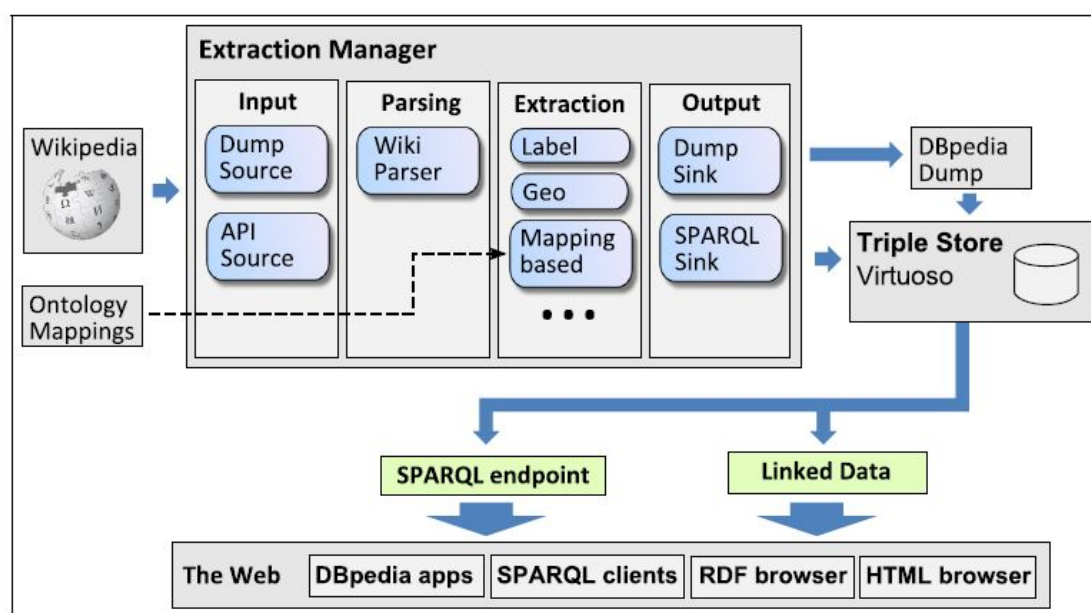


Figure 3.5: DBpedia Extraction Framework[10]

Such extractors are abstract, article categories, category label, category hierarchy, disambiguation, external links, geo coordinates, grammatical gender, homepage,

<sup>13</sup><https://lists.wikimedia.org/mailman/listinfo/wikidata-tech>

<sup>14</sup>[https://en.wikipedia.org/wiki/South\\_Pole\\_Telescope](https://en.wikipedia.org/wiki/South_Pole_Telescope)

image, infobox, interlanguage, label, lexicalizations, mappings, page ID, page links, persondata, PND, redirects, revision ID, thematic concept, topic signatures and wiki page.

DBpedia ontology<sup>15</sup> has 765 classes and 63152 properties[55]. DBpedia class is of three sections: labels, class hierarchy and properties. Labels section contains the class labels in different languages, whereas class hierarchy maps to the super class. All class properties are listed in the properties section. This section also contains property name, labels, domain, range and comment. A snapshot of the University class<sup>16</sup> is shown in figure 3.6.

<b>University</b> ( <a href="#">Show in class hierarchy</a> )				
<b>Label (es):</b> universidad <b>Label (pl):</b> uniwersytet <b>Label (ko):</b> 대학 <b>Label (el):</b> πανεπιστήμιο <b>Label (ga):</b> ollscoil <b>Label (fr):</b> université <b>Label (en):</b> university <b>Label (de):</b> Universität <b>Label (ja):</b> 大学 <b>Label (pt):</b> universidade <b>Label (nl):</b> universiteit <b>Super classes:</b> <a href="#">EducationalInstitution</a>				
<b>Properties on <i>University</i>:</b>				
Name	Label	Domain	Range	Comment
<a href="#">alumni</a> ( <a href="#">edit</a> )	alumni	<a href="#">EducationalInstitution</a>	<a href="#">Person</a>	
<a href="#">athletics</a> ( <a href="#">edit</a> )	athletics	<a href="#">University</a>	<a href="#">owl:Thing</a>	
<a href="#">campus</a> ( <a href="#">edit</a> )	campus	<a href="#">University</a>	<a href="#">owl:Thing</a>	

Figure 3.6: University Class in DBpedia

Table 3.9 shows some of Birzeit University entity properties in DBpedia.

<sup>15</sup><http://mappings.dbpedia.org/server/ontology/classes/>

<sup>16</sup><http://mappings.dbpedia.org/server/ontology/classes/University>

Table 3.9: Birzeit University Entity Properties in DBpedia

Property	Value
rdf:type	yago:EducationalInstitution108276342
rdf:type	schema:EducationalOrganization
dct:subject	Birzeit University
Wikipage page ID	2247791
Wikipage revision ID	683879742
Link from a Wikipage to...	<a href="http://www.birzeit.edu">http://www.birzeit.edu</a>
Link from a Wikipage to...	<a href="http://www.fobzu.org/">http://www.fobzu.org/</a>
affiliations	Mediterranean University Union
city	Birzeit
country	State of Palestine
established	1924
logo	a link
Motto	Building a Better Palestinian Future
rdfs:label	Birzeit University
rdfs:label	جامعة بيرزيت

Each property has its own page which contains all of its elements, such as labels in different languages, comments in different languages, domain, range, type, subPropertyOf, equivalentProperty and propertyDisjointWith.

DBpedia maps Wikipedia templates into classes. For Arabic, table 3.10 shows the status of templates mapping in Arabic. DBpedia has two endpoints, public

Table 3.10: Arabic Mapping of Classes and Properties in DBpedia

	Number	Percentage	#Occurrences	%Occurrences
Templates	40 of 1469	2.72	67427 of 269320	25.04 %
Proprieties	771 of 64902	1.19 %	333531 of 3710849	8.99 %

static endpoint<sup>17</sup> and live one<sup>18</sup>. The first one uses most of the datasets from DBpedia downloads, whereas the second one uses the most updated data from Wikipedia, where data sometimes is updated at least every one hour. We tested the

<sup>17</sup><http://dbpedia.org/sparql>

<sup>18</sup> <http://live.dbpedia.org/sparql>

updating mechanism for the live endpoint by updating some articles in Wikipedia and monitoring their correspondences in DBpedia but it didn't work. As per a discussion in DBpedia Discussion list<sup>19</sup> on May 10, 2016, DBpedia live extractor is stalled due to a change in the Wikipedia update stream, and the extractor maintenance also depends on the article language chapter.

Table below shows a comparison between the three knowledge bases described above.

Table 3.11: A Comparison between YAGO, Wikidata and DBpedia

	<b>YAGO</b>	<b>Wikidata</b>	<b>DBpedia</b>
Language	Properties are available only in English. Data is multilingual	Properties are in English mainly but some of them are available in other languages. Data is multilingual	Main Language is English. But linked to local chapters
Data Model	RDF Triples with Time and Location	Statements and Properties	Triples
Data Format	RDF TTL and TSV	JSON, XML, SQL, and RDFa	RDF (nt, nq, ttl)
Query Language	SPARQL	Wikibase-API, SPARQL	SPARQL
Source of Knowledge	Wikipedia, WordNet and GeoNames	Wikipedia and community effort	Wikipedia
External vocabulary	RDFS, OWL, WordNet	Not available, but sometimes equivalent properties are mapped	FOAF, RDFS, OWL, YAGO, UMBEL, Schema
Continuous Update	-	Yes, community effort	Yes, extraction from Wikipedia and community effort

After reviewing the three main knowledge bases above, the following points can be concluded:

- The Arabic structured data in these knowledge bases either only accessible via English relations, like the case in YAGO, or very small, like the case of Wikidata and DBpedia
- The inter-sources links can be useful to find missing or unavailable relations in data retrieval

<sup>19</sup><https://lists.sourceforge.net/lists/listinfo/dbpedia-discussion>

- Translate relations and name transliteration are mandatory steps to make Arabic contents accessible

In order to make use of the foreign structured data, the first part of the proposed approach depends on finding the properties of the user's query in all available knowledge bases. If the Arabic property is found in a system, then one can query that system with this property. If not, the inter-resources links can be used to find the property syn-set, then the query expansion can be applied. In case a conflict or inconsistency is found in the retrieved data from different sources, the percentage of certainty can be displayed to user along with results.

This approach will be flexible enough to make use of any new knowledge base in the future. It also doesn't eliminate the possibility to make use of the web semi- or un-structured data to extract knowledge.

### 3.2.2 Generate Arabic Structured Data from Foreign Language Structured Data

Since Wikipedia is one of the main sources for YAGO facts, one can make use of the cross lingual links in Wikipedia and Wikidata to translate YAGO facts to Arabic. To do that, facts need to be parsed first, in which property, subject and object identified. Then, each of them needs to be translated apart then combined to generate the Arabic fact.

**Fact Parser** YAGO Facts can be downloaded as Tab Separated file (TSV). We downloaded sample file<sup>20</sup>, and had to develop a simple parser that parses the fact file into array of facts objects, In order to make use of the foreign structured data, the

rst part of the proposed as follows:

- Remove the extra tab at the beginning of each fact
- Remove the surrounding <and >that surrounds each of the subject, property and object
- Replace the underscore by space in each subject and object

The fact object would contain subject, property, object and language.

---

<sup>20</sup><https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/>

**Using Wikipedia and Wikidata in Translation and Transliteration** Before going into the details of translation, the concept of making use of the cross-lingual links in Wikipedia and Wikidata in translation and transliteration needs to be revisited. When we have a title in a language, say English, and it has a Wikipedia article in the English Wikipedia, one can access the corresponding article in another Wikipedia, like Arabic, using the Wikibase-Item of this article. This can be done by retrieving the article page, like what has been done in parsing Wikipedia articles, then get Wikibase-Item from that page. After that, the Wikidata item of that Wikibase-Item is retrieved using Wikidata API. The item data members to retrieve are specified in the API request, and one can retrieve the multilingual members, such as label, aliases and descriptions. In such way, the Arabic title of that Wikipedia article can be retrieved as Wikidata label or aliases. Even the Arabic descriptions of that English title can be retrieved in the same way as well.

To retrieve the Wikipedia article page of a title, Wikipedia API can be used, and to retrieve the Wikidata item of the article Wikibase-Item, Wikidata API is used. Request 3.17 below is used to retrieve the Wikidata item for Wikibase-Item “Q510029”, which is for “Birzeit University” article.

```
https://www.wikidata.org/w/api.php?action=wbgetentities&format=xml&languages=
ar&props=aliases|labels|descriptions|claims&ids=Q510029
```

(3.17)

The XML snippet 3.2.2 below shows part of the response:

```
<api success="1">
<entities>
  <entity type="item" id="Q510029">
    <labels>
      <label language="en" value="Birzeit University" />
    </labels>
    <descriptions>
      <description language="en" value="university" />
    </descriptions>
    <aliases />
    <claims>
      <property id="P31">
        <claim type="statement" id="Q510029$40b1d4e4-4d2d-b78a-98
        ↪ af-b0d39cce34ec" rank="normal">
          <main snak snaktype="value" property="P31" hash="482
          ↪ db304b57e4b3a7a5b3a6e2256db7c639c49b0" datatype="
          ↪ wikibase-item">
```

```

    <datavalue type="wikibase-entityid">
      <value entity-type="item" numeric-id="3918" id="Q3918
        ↪ " />
    </datavalue>
  </mainsnak>
</claim>
</property>
.
.
.

```

XML parser has been developed to parse this response and extract the data members of the Wikidata item.

What if no Wikipedia article is found for that title? One can make use of the Wikidata Sparql endpoint to query Wikidata for items by label or alias. The Sparql query 3.2.2 retrieves all Wikidata items with label “Birzeit University”.

```

SELECT DISTINCT ?item WHERE {
  ?item ?label "Birzeit University"@en.
  ?article schema:about ?item.
  ?item rdfs:label ?itemLabel.
  FILTER("Birzeit University"@en = ?itemLabel)
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en
    ↪ ". }
}

```

Query can be executed either manually in the Sparql Endpoint or using the Wikidata API, as in request 3.18.

```

https://query.wikidata.org/sparql?query=SELECTDISTINCT?itemWHERE{?item?
label"BirzeitUniversity"@en.?articleschema:about?item.?itemrdfs:label?
itemLabel.FILTER("BirzeitUniversity"@en=?itemLabel)SERVICEwikibase:label{bd:
serviceParamwikibase:language"en".}}

```

(3.18)

The XML response 3.2.2 is retrieved for the request above:

```

<?xml version='1.0' encoding='UTF-8'?>
  <sparql xmlns='http://www.w3.org/2005/sparql-results#'>
    <head>
      <variable name='item' />
    </head>

```



```

<results>
  <result>
    <binding name='item'>
      <uri>http://www.wikidata.org/entity/Q510029</uri>
    </binding>
  </result>
</results>
</sparql>

```

To extract the Wikibase-Item “Q510029” from this response, a simple XML parse has been developed. Similar query has been written to retrieve Wikidata item by alias.

The label and aliases values used in these queries are case sensitive. So some items might not be retrieved because they have different letter-cases. We tried different case insensitive queries to overcome this issue but all of them failed with timeout error. To solve this issue, query is invoked with different cases of the label and aliases then responses are combined.

The translateWikiTitle procedure (will be used later in translate YAGO subject and object) is proposed in 1 algorithm.

---

**Algorithm 1:** Translate Wiki Title

---

```

Input: title
Output: Translated Title

translatedTitles ← [ ]
/* Use Wikipedia procedure translateTitle */
translatedTitles ← translateWikipediaTitle(subject)
if translatedTitles is not empty then
  | /* Title is translated as Wikipedia Article title */
  | return translatedTitles
end
else
  | /* Couldn't translate title as Wikipedia Article Title. Try
  |   to translate it as Wikipedia Item */
  | translatedTitles ← translateWikidataTitle()
end
return translatedTitles

```

---

This algorithm tries to translate title as Wikiedia article title, which is implemented in algorithm 2. If no translation is found, title is translated as Wikidata title, as implemented in algorithm 3.

---

**Algorithm 2:** Translate Wikipedia Article Title

---

**Input:** title, description, aliasClasses, language, targetLanguage

**Output:** Title with Translated Descriptions

```
translatedTitle  $\leftarrow$  null
items  $\leftarrow$  [ ]

// Retrieve Wikipedia Items by title
wikipediaArticleItem  $\leftarrow$  retrieveWikipediaArticleItems()
if wikipediaArticleItem  $\neq$  null then
    /* Wikiedia article with that title found, and Wikidata Item
       for this article is retrieved as well */

    /* Get the title for that article in target language */
    translatedTitle  $\leftarrow$  wikipediaArticleItem.getLabel(targetLanguage) if
        translatedTitle  $\neq$  "" then
        /* Title in the target language for this Wikipedia
           article is found as well */

    end
    else
        /* Couldn't find Title in the target language for this
           Wikipedia artilce. Maybe this article doesn't exist
           in the target language */

    end
end
else
    /* No Wikipedia article with this title found */

end
return translatedTitle
```

---

---

**Algorithm 3:** Translate Wikidata Title

---

**Input:** title, description, titleClasses, language, targetLanguage

**Output:** Translated Title

```
/* Try to translate title as Wikidata Item Label */
translatedLabels ← translateTitleAsWikidataLabel()
if translatedLabels is not empty then
    /* Title is translated as Wikidata Item Label */
    return translatedLabels
end
else
    /* Couldn't translate Title as Wikidata Label. Try to
       translate it as Wikidata Item Alias */
    translatedAliases ← translateTitleAsWikidataAlias()
    if translatedAliases is not empty then
        /* Title is translated as Wikidata Item Alias */
        return translatedAliases
    end
    else
        /* Couldn't translate Title as Wikidata Alias. Try to
           append Wikidata Description in the target language */
        translatedDescriptions ← translateTitleAsWikidataDescription()
        if translatedDescriptions is not empty then
            /* Title is translated as Wikidata Item Description */
            return translatedDescriptions
        end
        else
            /* Couldn't translate Title as Wikidata Description */
            end
        end
    end
end
return title
```

---

When trying to translate title as Wikidata title, Wikidata items are being queried by label, then those retrieved items with label available in the Arabic are returned, as implemented in algorithm 4. If no item is found, Wikidata items are being queried by alias, then those retrieved items with label or aliases available in Arabic are returned, as implemented in algorithm 5. If no item is found, those

items retrieved as Wikipedia items or Wikidata items when querying Wikidata by label or alias but have no title or alias in Arabic are visited again, and the Arabic descriptions of them are appended to the English title. This is implemented in algorithm 6.

---

**Algorithm 4:** Translate Wikidata Item Label

---

**Input:** title, description, aliasClasses, language, targetLanguage

**Output:** Translated Labels

```

translatedLabels  $\leftarrow$  null
/* Retrieve all Wikidata Items by label */
wikidataItems  $\leftarrow$  getWikidataItemsByLabel()
if wikidataItems is not empty then
    /* Some Wikidata Items in the source language found by label
    */
    /* Loop over wikidataItem to check if they have labels in
    target language */
    foreach item in wikidataItems do
        translatedLabel  $\leftarrow$  item.getLabel(targetLanguage)
        if translatedLabel  $\neq$  "" then
            /* Label in target language found */
            translatedLabels.add(translatedLabel)
        end
        else
            /* Label in target language is not found */
        end
    end
    return translatedLabels
end
else
    /* Couldn't find Wikidata Item in source language by label
    */
end
return translatedLabels

```

---

---

**Algorithm 5:** Translate Wikidata Item Alias

---

**Input:** title, description, aliasClasses, language, targetLanguage

**Output:** Translated Aliases

```
translatedAliases ← null
/* Retrieve all Wikidata Items by alias */
wikidataItems ← getWikidataItemsByAlias()
if wikidataItems is not empty then
    /* Some Wikidata Items in the source language found by
       alias. Loop over wikidataItem to check if they have
       aliass in target language */
    foreach item in wikidataItems do
        translatedAlias ← item.getLabel(targetLanguage)
        if translatedAlias ≠ "" then
            /* Label in target language found */
            translatedAliases.add(translatedAlias)
        end
        else
            /* Label in target language is not found. Check for
               alias in target language */
            translatedAlias ← item.getAlias(targetLanguage) if
            translatedAlias ≠ "" then
                /* Alias in target language found */
                translatedAliases.add(translatedAlias)
            end
            else
                /* Alias in target language is not found */
            end
        end
    end
end
else
    /* Couldn't find Wikidata Item in source language by alias
       */
end
return translatedAliases
```

---

---

**Algorithm 6:** Translate Wikidata Item Description

---

**Input:** title, description, aliasClasses, language, targetLanguage

**Output:** Title with Translated Descriptions

translatedDescriptions  $\leftarrow$  null

items  $\leftarrow$  [ ]

// Retrieve all Wikipedia Items by title

wikipediaItem  $\leftarrow$  retrieveWikipediaArticleItems()

items.add(wikipediaItem)

// Retrieve all Wikidata Items by label

wikidataItems  $\leftarrow$  retrieveWikidataItemsByLabel()

items.add(wikidataItems)

// Retrieve all Wikidata Items by alias

wikidataItems  $\leftarrow$  retrieveWikidataItemsByAlias()

items.add(wikidataItems)

**if** *items is not empty* **then**

    /\* Some Wikidata Items in the source language found by  
    Wikipedia title, Wikidata Item Label and Alias. Loop  
    over Items to check if they have descriptions in target  
    language \*/

**foreach** *item in items* **do**

        itemTranslatedDescriptions  $\leftarrow$  item.getDescription(targetLanguage)

**if** *itemTranslatedDescriptions is not empty* **then**

            /\* Descriptions in target language found \*/

            translatedDescriptions.addAll(itemTranslatedDescriptions)

**end**

**else**

            /\* Description in target language is not found \*/

**end**

**end**

    /\* Append found descriptions in the target language to the  
    title in the source language \*/

    translatedTitle = title + "(" + translatedDescriptions + ")"

**end**

**else**

    /\* Couldn't find Wikidata Item in source language by alias  
    \*/

**end**

**return** translatedTitle

---

**Translate Properties** Since there are only 102 English properties in YAGO, we translated them manually to Arabic.

**Translate Subject** We found that most of the subjects are titles for Wikipedia articles or Wikidata items. Translate Wiki Title algorithm 1 is used to translate YAGO subject. Algorithm 7 shows the detailed procedure.

---

**Algorithm 7:** Translate YAGO Subject

---

**Input:** subject: Fact Subject

**Output:** Translated Subject

translatedSubjects  $\leftarrow$  [ ]

*/\* Use Wikipedia and Wikidata procedures to translate subject  
as title \*/*

translatedSubjects  $\leftarrow$  translateWikiTitle(subject)

**if** *translatedSubjects is not empty* **then**

*/\* Title is translated as Wikipedia Article title \*/*  
    **return** translatedSubjects

**end**

**else**

*/\* Couldn't translate label as Wiki Article Title \*/*  
    translatedSubjects = [subject]  
    **return** translatedSubjects

**end**

---

**Translate Object** Similar to subject translation, YAGO objects are translated using Wikipedia and Wikidata titles at first, then if no translation found, Google Translate API is used, as shown in algorithm 8.

---

**Algorithm 8:** Translate YAGO Object

---

**Input:** object: Fact Subject

**Output:** Translated Object

```
translatedObjects ← null
/* Try to translate Object using Wiki */
translatedObjects ← translateObjectUsingWiki(object)
if translatedObjects is empty then
    | /* Object is translated using Wiki */
    |
end
else
    | /* Object couldn't be translated using Wiki. Translate it
      | using Google Translate API */
    | translatedObjects ← translateObjectUsingGoogleTranslate(object)
end
return translatedObjects
```

---

However, initial results showed that number of translated objects in this way is much less than translated subjects. We studied the translated objects and found that:

1. Objects sometimes have extra details, such as description that is appended to the end of the object between brackets or a related object which is appended to the end of the object by a comma. Such details usually are not part of the Wikipedia or Wikidata title. If the description is deleted, then the remaining object can be translated using algorithm 1. For the second case, object can be split into sub-objects and each of them can be translated using algorithm 1, then can be combined together to have the completely translated objects.
2. Some objects can't be translated using Wikipedia or Wikidata items, but they contain phrases that match specific patterns. The "key" phrase usually can be translated using Wikipedia or Wikidata. "Battle of Harpers Ferry" is an example of these objects. This object can't be translated using algorithm 1, but if it can be split into "Battle of" and "Harpers Ferry", then algorithm 1 can work on the second part. The first part can be translated manually.
3. Some objects (and subjects) represent several concepts, and meaning can be distinguished only in the context. Translating these objects to Arabic may produce several concepts, where only one of them is correct. For example, the fact <Palestine><hasNeighbour><Lebanon> was translated to the following facts:



- (a) <لبنان، مقاطعة واويكا><له جار><فلسطين>
- (b) <ليبانون><له جار><فلسطين>
- (c) <لبنان، اوهايو><له جار><فلسطين>
- (d) <لبنان><له جار><فلسطين>
- (e) <لبنان، مقاطعة واوباكا><له جار><فلسطين>

Only the third fact has a correct object, the other's objects are for states and villages. And the same for the fact <India><hasNeighbour><Nepal>, which was translated to the following facts:

- (a) <نيبال><له جار><إنديا>
- (b) <نيبال><له جار><الهند>

The object of the second translated fact is a cat name.

The fact <Australia><hasCapital><Canberra> was translated to the following two facts:

- (a) <كانبرا><له عاصمة><أستراليا>
- (b) <إنجلش إلكتروك كانبيرا><له عاصمة><أستراليا>

The object of the first translated fact is the correct one (capital), the other one is a name of an aircraft family.

To solve this issue, subject and object types or classes need to be taken into consideration when translating. i.e if the translated subject or object matches the expected fact subject or object type, accepts it, otherwise rejects it.

**Object Map** To dynamically support the different object patterns, object map was implemented. It contains the possible object patterns as Java standard regex and their replacement. The XML snippet 3.2.2 below is part of the object map.

```

<MappedObjects>
  <MappedObject object="wordnet ([\p{L}['-]\s+)" replacement
    ↪ ="[OBJECT=1]"/>
  <MappedObject object="Battle of ([\p{L}['-]\s+)" replacement=" [
    ↪ OBJECT=1]"/>
  <MappedObject object="Battle for ([\p{L}['-]\s+)" replacement="
    ↪ [OBJECT=1]"/>
  <MappedObject object="Attack on ([\p{L}['-]\s+)" replacement="
    ↪ [OBJECT=1]"/>
  <MappedObject object="(\d+) Attack on ([\p{L}['-]\s+)"
    ↪ replacement=" [OBJECT=2] [1]"/>
  <MappedObject object="Air battle of ([\p{L}['-]\s+)" replacement
    ↪ =" [OBJECT=1]"/>
  <MappedObject object="Naval battles of the ([\p{L}['-]\s+)"
    ↪ replacement=" [OBJECT=1]"/>
  <MappedObject object="(\d+) in the ([\p{L}['-]\s+)" War"
    ↪ replacement=" [OBJECT=2] [1]"/>
  .
  .

```

The fact object “Battle of Harpers Ferry” mentioned above, matches the object pattern “Battle of ([\p{L}['-]\s+)”. If a Java regex matcher is used to match the regex with the object, two matching groups are found: [0] = “Battle of Harpers Ferry” and [1] = “Harpers Ferry”. The index of the group that needs translation is 1 (0-based index), which is specified in [OBJECT=1] in the replacement. So “Harpers Ferry” will be translated. The translation tool will translate this part and append it to “معركة”, which is also specified in the replacement part.

**Property Map** To add constraints to the classes of the translated subject and object, we implemented property map. Subject and Object classes can be retrieved from Wikidata, where each item has a statement called “isInstanceOf” that specifies the item class, which is also a Wikidata item. To specify the expected subject and object classes for an YAGO property, we looked for the corresponding property in Wikidata and matched the property constraints, which usually specify the expected classes of the property domain and range. This is not straightforward, sometimes these constraints are not well defined or don’t include all the classes, and it is needed to scan YAGO properties subjects and objects, find them in Wikidata and collect their classes. We built an XML map file that has all these details in addition to the Arabic translation of the property. The XML snippet below shows part of the property map file.

```

<MappedProperty yAGOProperty="hasCapital" translatedProperty=" ">

```

```

<WikidataProperty wikibaseItem="" label="capital">
  <SubjectClasses>
    <SubjectClass wikibaseItem="Q6256" label="country" status="
      ↪ INCLUDE"/>
    <SubjectClass wikibaseItem="Q486972" label="human settlement"
      ↪ status="INCLUDE"/>
    <SubjectClass wikibaseItem="Q23442" label="island" status="
      ↪ INCLUDE"/>
    <SubjectClass wikibaseItem="Q133442" label="city-state" status="
      ↪ INCLUDE"/>
    <SubjectClass wikibaseItem="Q3373417" label="country of origin"
      ↪ status="INCLUDE"/>
    <SubjectClass wikibaseItem="Q56061" label="administrative
      ↪ territorial entity" status="INCLUDE"/>
    <SubjectClass wikibaseItem="Q82794" label="geographic region"
      ↪ status="INCLUDE"/>
    <SubjectClass wikibaseItem="Q3024240" label="historical country"
      ↪ status="INCLUDE"/>
    <SubjectClass wikibaseItem="Q512187" label="federal republic"
      ↪ status="INCLUDE"/>
    <SubjectClass wikibaseItem="Q48349" label="empire" status="
      ↪ INCLUDE"/>
    <SubjectClass wikibaseItem="Q7275" label="state" status="INCLUDE
      ↪ "/>
    <SubjectClass wikibaseItem="Q7270" label="republic" status="
      ↪ INCLUDE"/>
    <SubjectClass wikibaseItem="Q3624078" label="sovereign state"
      ↪ status="INCLUDE"/>
    <SubjectClass wikibaseItem="Q1620908" label="historical region"
      ↪ status="INCLUDE"/>
    <SubjectClass wikibaseItem="Q3502482" label="cultural region"
      ↪ status="INCLUDE"/>
    <SubjectClass wikibaseItem="Q28171280" label="ancient
      ↪ civilization" status="INCLUDE"/>
    <SubjectClass wikibaseItem="Q4167410" label="Wikimedia
      ↪ disambiguation page" status="EXCLUDE"/>
    <SubjectClass wikibaseItem="Q4167836" label="Wikimedia category"
      ↪ status="EXCLUDE"/>
    <SubjectClass wikibaseItem="Q13406463" label="Wikimedia list
      ↪ article" status="EXCLUDE"/>

```

```

    <SubjectClass wikibaseItem="Q17329259" label="encyclopedic
        ↪ article" status="EXCLUDE"/>
</SubjectClasses>
<ObjectClasses>
    <ObjectClass wikibaseItem="Q515" label="city" status="INCLUDE"/>
    <ObjectClass wikibaseItem="Q5119" label="capital" status="
        ↪ INCLUDE"/>
    <ObjectClass wikibaseItem="Q2264924" label="port city" status="
        ↪ INCLUDE"/>
    <ObjectClass wikibaseItem="Q5715" label="ancient city" status="
        ↪ INCLUDE"/>
    <ObjectClass wikibaseItem="Q1549591" label="big city" status="
        ↪ INCLUDE"/>
    <ObjectClass wikibaseItem="Q4167410" label="Wikimedia
        ↪ disambiguation page" status="EXCLUDE"/>
    <ObjectClass wikibaseItem="Q4167836" label="Wikimedia category"
        ↪ status="EXCLUDE"/>
    <ObjectClass wikibaseItem="Q13406463" label="Wikimedia list
        ↪ article" status="EXCLUDE"/>
    <ObjectClass wikibaseItem="Q17329259" label="encyclopedic
        ↪ article" status="EXCLUDE"/>
</ObjectClasses>
</WikidataProperty>
</MappedProperty>

```

When retrieving the items for Wikipedia title or from Wikidata by label or alias, the item class is validated against property subject or object classes. If it matches one of the classes, and the class status is “INCLUDE”, then this item is accepted as translation. If the status is “EXCLUDE”, then the item is rejected. Wildcards, such as \* are accepted as well. Back to the translated facts of the fact <Australia><hasCapital ><Canberra >, only the translated object **كانبرا** is accepted because its Wikidata item (Q520964) is instance of “capital” and “city”, which are part of the included subject classes. The other translated object **إِنجَلش إلكترىك كانبيرا** is rejected, because its Wikidata item (Q520964) is instance of “aircraft family”, which is not listed in the included object classes.

---

**Algorithm 9:** Translate YAGO Object Using Wiki

---

**Input:** object: Fact Subject

**Output:** Translated Object

```
translatedObjects ← []
/* Translate Object as Wiki Title */
translatedObjects ← translateWikiTitle(object)
if translatedObjects is not empty then
    /* Title is translated as Wiki Article title */
    return translatedObjects
end
else
    /* Try to match the object with one of the Object regex */
    matchedObject ← match(Object)
    if matchedObject ≠ object then
        /* Object matched one of the Object defined regex */
        translatedObjects ← translateWikiTitle(matchedObject)
        if translatedObjects is not empty then
            /* Replace the translated object part in the original
            */
            translatedObjects ← replaceObject(object, translatedObjects)
        end
    end
    else
        /* Check if object contains description or comma */
        if object.lastChar() = “,” then
            /* Object contains description */
            cleanObject ← removeDescription(object)
            translatedObjects ← translateWikiTitle(cleanObject)
        end
        else if object.contains(“,”) then
            /* Split it into parts, translate each part then
            combine them */
            subObjects ← object.split(“,”)
            foreach subObject in subObjects do
                translatedSubObjects ← translateWikiTitle(subObject)
                translatedObjects.addAll(translatedSubObjects)
            end
            translatedObjects ← combineSubjObjects(translatedObjects)
        end
    end
end
end
```

---

---

**Algorithm 10:** Translate YAGO Object Using Google Translate API

---

**Input:** object: Fact Subject

**Output:** Translated Object

```
translatedObjects  $\leftarrow$  []  
/* Try to match the object with one of the Object regex */  
matchedObject  $\leftarrow$  match(Object)  
if matchedObject  $\neq$  object then  
    /* Object matched one of the Object defined regex.  
       Translate the matched object using Google Translate API  
       */  
    translatedObjects  $\leftarrow$  translateUsingGoogle(matchedObject)  
    translatedObjects  $\leftarrow$  replaceObject(object, translatedObjects)  
end  
else  
    /* Check if object contains description or comma */  
    if object.lastChar() = “,” then  
        /* Object contains description */  
        cleanObject  $\leftarrow$  removeDescription(object)  
        description  $\leftarrow$  getDescription(object)  
        translatedObject  $\leftarrow$  translateUsingGoogle(cleanObject)  
        translatedDescription  $\leftarrow$  translateUsingGoogle(description)  
        translatedObject =  
            translatedObject + “(” + translatedDescription + “)”  
        translatedObjects = [translatedObject]  
    end  
    else if object.contains(“,”) then  
        /* Split it into parts, translate each part then combine  
           them */  
        subObjects  $\leftarrow$  object.split(“,”)  
        foreach subObject in subObjects do  
            translatedSubObject  $\leftarrow$  translateUsingGoogle(subObject)  
            translatedObjects.add(translatedSubObject)  
        end  
        translatedObjects  $\leftarrow$  combineSubjObjects(translatedObjects)  
    end  
    else  
        /* Object didn't match any of the patterns. Translate is  
           as it is */  
        translatedObject  $\leftarrow$  translateUsingGoogle(object)  
        translatedObjects = [translatedObject]  
    end  
end  
return translatedObjects
```

---

## Generating Arabic Unstructured Data from Foreign Language Structured Data

Wikipedia 2016 statistics showed that there are 425,406 articles in the Arabic Wikipedia, which is about 1.07% of the total articles and about 8.2% of the articles in the English Wikipedia [56]. However, these numbers may not reflect size of the useful Arabic articles, or even not the non short ones. We found that among the 425,406 articles in Arabic Wikipedia, there are 226,797 article (53.3%) that have less than 200 words. We have looked at some of these article could conclude the following:

- Most of these article are for cities, films, plants and animals
- Some of them have a csv and Excel file as a reference, which means their data were automatically generated
- They are short articles with less than 200 words
- Most of them (specially the cities) are for non Arabic items (such as towns in North America)
- Some of the articles' authors have more than 7000 articles. We could find that most of their articles have the same structure

So, the actual size of the Arabic content in the web seems to be much less than the statistics show.

Traditional method to enhance the Arabic content in web is to create articles. This approach can be automated. One can make use of the foreign language structured data to generate a qualitative articles in the following way:

- Analyze Wikipedia articles based on their categories, then generate a template for each category
- Use the system proposed in the previous section to query the knowledge bases for all the available properties of the article to be generated
- Fill the template with the structured data returned from the query system
- Create or update the article item in Wikidata
- Generate a Wikipedia article from the template
- Create an Infobox for the generated article and link it to Wikidata

Such article may be marked as “Auto Generated Article” in its Wikipedia page.

### 3.3 Expanding Work to Other Languages

The proposed approach for cross lingual similarity can be expanded easily to new languages. This is already fully automated except the following two steps where user needs to update the following language-specific items:

- Identifying the language specific keywords in the category titles, like the ones we figured out when filtering out the representative categories, such as “تصنيف” in Arabic and “reference” in English, and when parsing the article data
- Using proper preprocessing algorithms like the the Lucene Stemmer and POS Tagger

In addition to the minor code changes related to support the new languages in Wikipedia API calls.

Expanding the YAGO fact translation is also fully automated. Minor code changes are needed to support the language code to retrieve Wikipedia and Wikidata data.



# Chapter 4

## Results

### 4.1 Cross Lingual Document Similarity

#### 4.1.1 Selecting Representative Category

To test the proposed approach; that is to find similar documents based on Wikipedia concepts; articles and categories, similarity tests were executed for documents from different sets. These sets are:

- Mutual Wikipedia Featured Articles. Those featured articles that exist in both Arabic and English Wikipedia
- Translated documented ranked by Birzeit University students
- Tweets
- Global Voices documents

The Wikibase-Item vector for each document was constructed using Wikipedia articles and categories, as concepts, so each document has two types of vectors. The category Wikibase-Item vector was constructed for each representative categories set, then the similarity tests were executed for them. For a given representative category set and independent of language, category vectors for all documents contain the same components but with different weights (weight might be equal to zero). The same for article vectors.

#### 4.1.2 Wikipedia Featured Articles

Featured articles are used as examples to write new articles, as they are some of the best articles in Wikipedia as determined by Wikipedia editors. Featured article candidates are evaluated for accuracy, neutrality, completeness and style.

There are 5354 featured articles in English Wikipedia ( 0.1%)[57] and 553 in the Arabic Wikipedia (0.95%) [58].

The Wikipedia Mutual Featured Articles test set was selected from the featured articles in Arabic and English in two steps:

- Get the common featured articles in Arabic and English by intersecting the featured articles Wikibase-Items from both languages
- Filter the common featured articles manually by removing those talk about very generic concepts, such as bird and Association football

After these two steps, we had 52 featured articles in each language, and are listed in appendix B. Similarity tests were executed for vectors representing text chunks or variable sizes as pairs of articles in the same language, pairs of articles in different languages, articles and articles introductions in the same language, articles and articles introductions in different languages and pairs of articles introductions in the same language.

The article introduction represents a summary for the article topic, and it is the lead section that prepares the reader for the detail in the subsequent sections [57]. Table 4.1 shows the similarity test results for queries of Arabic Articles vs English Articles, Arabic Introductions and English Introductions.

Table 4.1: Similarity Test Results for Queries of Arabic Articles against Collections of Different Types and Languages

<b>Vector/Collect. Type</b>	<b>English Articles</b>	<b>Arabic Introductions</b>	<b>English Introductions</b>
47012 Categories	<45,34,12>	<50,50, <u>46</u> >	<50,40,22>
24825 Categories	<51,47, <u>28</u> >	<50,50,41>	< <u>52</u> ,45,29>
19031 Categories	<51,47,27>	<50,50,40>	< <u>52</u> , <u>49</u> ,28>
13786 Categories	<51, <u>48</u> ,26>	<50,50,37>	< <u>52</u> ,47,27>
12076 Categories	<51,46,23>	<50,50,36>	<50,48,25>
9178 Categories	< <u>52</u> ,47,23>	<50,50,32>	<51, <u>49</u> , <u>32</u> >
Articles	<43,33,14>	<50,47,26>	<47,35,15>

The 3-tuple vector in each column represents the number of queries when its corresponding document is found in the top 10, top 5 and first results, respectively.

Table 4.1 shows that using category vectors always gives better results than using article vector. The best results are obtained when using:

- **24825 Categories** to have corresponding English document as the first retrieved document

- **13786 Categories** to have corresponding English document in the first 5 retrieved documents
- **9178 Categories** to have corresponding English document in the first 10 retrieved documents

When executing the similarity tests for Arabic articles as queries against a collection of Arabic articles introductions, changing the representative category set had no effect on the results except the results of having the corresponding article introduction as the first retrieved document. For this test, accuracy decreases as number of representative categories decreases. When having Arabic articles as queries but against a collection of English articles introductions. Changing the representative category set significantly affects accuracy. Table 4.1 shows that best results are obtained when using:

- **9178 Categories** to have corresponding English document as the first retrieved document
- **19031 Categories** or **9178 Categories** to have corresponding English document in the first 5 retrieved documents
- **24825 Categories**, **19031 Categories** or **13786 Categories** to have corresponding English document in the first 10 retrieved documents

Queries of English articles against collections of Arabic articles, English introductions and Arabic introductions were tested next. Table 4.2 shows the results.

Table 4.2: Similarity Test Results for Queries of English Articles and Documents of Different Collections

Vector/Collect. Type	Arabic Introductions	English Introductions
47012 Categories	<38,29,13>	<52,52,48>
24825 Categories	< <u>48</u> ,40, <u>24</u> >	<52,52,49>
19031 Categories	<47,40,23>	<52,52,49>
13786 Categories	<47, <u>41</u> ,20>	<52,52,49>
12076 Categories	<47, <u>41</u> ,20>	<52,52,49>
9178 Categories	<47,40,12>	<52,52,48>
Articles	<39,28,9>	<52,52,44>

One can see that best results for the queries of English articles and a collection of Arabic Introductions are obtained when using

- **24825 Categories** to have corresponding English document as first retrieved document
- **13786 Categories** or **12076 Categories** to have corresponding English document in the first 5 retrieved documents
- **24825 Categories** to have corresponding English document in the first 10 retrieved documents

However, for the collection of English article introductions, changing the representative category set doesn't affect number of retrieved documents with rank  $\leq 10$ ,  $\leq 5$  or  $= 1$ .

Test queries of Arabic article introductions against a collection of English article introductions comes last. Table 4.3 shows that changing the representative category set doesn't significantly affect number of retrieved documents with rank  $\leq 10$  or with rank  $\leq 5$ , and best results are achieved when using 19031 Categories to get the most similar document at the top of the retrieved collection.

Table 4.3: Similarity Test Results for Queries of Arabic Articles Introductions and a Collection of English Article Introductions

Vector/Collect. Type	English Introductions
47012 Categories	<49,43,30>
24825 Categories	<51, <u>51</u> ,39>
19031 Categories	<51,50, <u>41</u> >
13786 Categories	<51,50,39>
12076 Categories	<51,50,38>
9178 Categories	<50,50,38>
Articles	<50,49,31>

### 4.1.3 Translated News Articles

The next test set is a collection of 46 medium and long translated news articles in Arabic and English. These articles were collected from the Web by engineering students at Birzeit University. The article titles in Arabic and English are listed in appendix C. Similarity tests were done by finding the similarity between each article and the collection articles (the queries and the collection documents here are the same). Results are shown in Table 4.4

Table 4.4: Similarity Test Results for Translated News Articles

Vector/Collect. Type	Result Vector
47012 Categories	<34,28,11>
24825 Categories	<42,41,28>
19031 Categories	<43,41,32>
13786 Categories	<43,42,32>
12076 Categories	<44,42,32>
9178 Categories	< <u>45</u> ,42, <u>33</u> >
Articles	<41,39,27>

Similarity test results show that best results can be obtained when using 9178 categories to generate category vectors.

#### 4.1.4 Bilingual Tweets

Similarity tests then were executed for a collection of Tweets in Arabic and English. A total of 128 Arabic and English Tweets were collected from official and non-official accounts. Tweets are considered to be short documents, as most of them are less than 144 characters in length.

Table 4.5 shows the similarity test results:

Table 4.5: Similarity Test Results of Bilingual Tweets

Vector/Collect. Type	Result Vector
47012 Categories	<38,29,10>
24825 Categories	<58,51,32>
19031 Categories	<57,54,33>
13786 Categories	<57, <u>55</u> ,38>
12076 Categories	<58,53, <u>42</u> >
9178 Categories	< <u>59</u> ,53,39>
Articles	<57,55,37>

One can see that best results are obtained when using:

- **12076 Categories** to have corresponding English document as first retrieved document
- **13786 Categories** or **12076 Categories** to have corresponding English document in the first 5 retrieved documents

- **9178 Categories** to have corresponding English document in the first 10 retrieved documents

#### 4.1.5 Global Voices Parallel Corpus

Bilingual Documents from Global Voices Parallel Corpus then were used as input for similarity tests. Global Voices parallel corpus is a corpus of news stories in 38 languages and 74.85M tokens, that is customized by OPUS, the open parallel corpus [59].

Global Voices Parallel Corpus is available in XML, TMX (Translation Memory eXchange) and MOSES formats. For this research, TMS file was downloaded, then a parser was developed to convert it to a csv file that can be parsed by the similarity test application. The csv file then was filtered manually and total of medium-length 178 Arabic and English news documents were selected randomly. Similarity tests results are shown in table 4.6.

Table 4.6: Similarity Test Results of Global Voices Parallel Corpus

Vector/Collect. Type	Result Vector
47012 Categories	<56,45,12>
24825 Categories	<80,69,35>
19031 Categories	<88,47,38>
13786 Categories	<89,80,43>
12076 Categories	<88,81,46>
9178 Categories	<87,82,49>
Articles	<84,75,50>

One can see that best results are obtained when using:

- **Articles** then **9178 Categories** to have corresponding English document as first retrieved document
- **9178 Categories** to have corresponding English document in the first 5 retrieved documents
- **13786 Categories** to have corresponding English document in the first 10 retrieved documents

To get a better comparison picture for the results, percentages of documents whose corresponding documents are found in the first 10, 5 and 1 result documents are visualized.

Figures 4.1, 4.2, 4.3 and 4.4 show percentages of documents whose corresponding documents are found in the first 10 result documents in all similarity tests.

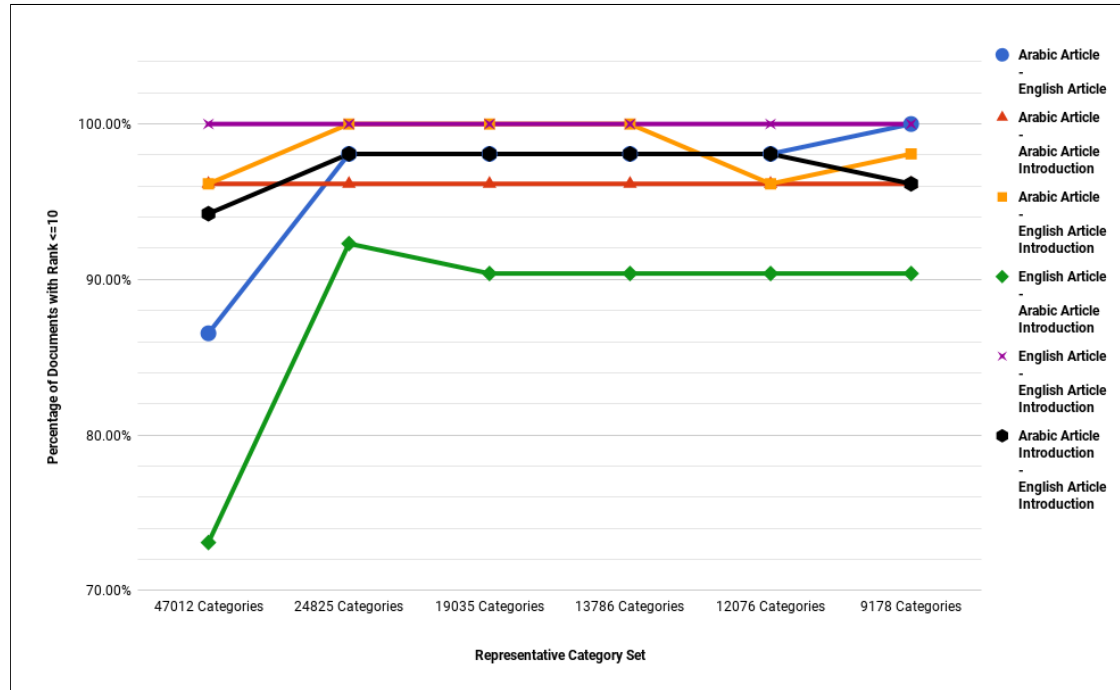


Figure 4.1: Percentage of Documents with Rank  $\leq 10$  to the Total number of Documents in the Collection for Wikipedia Featured Articles

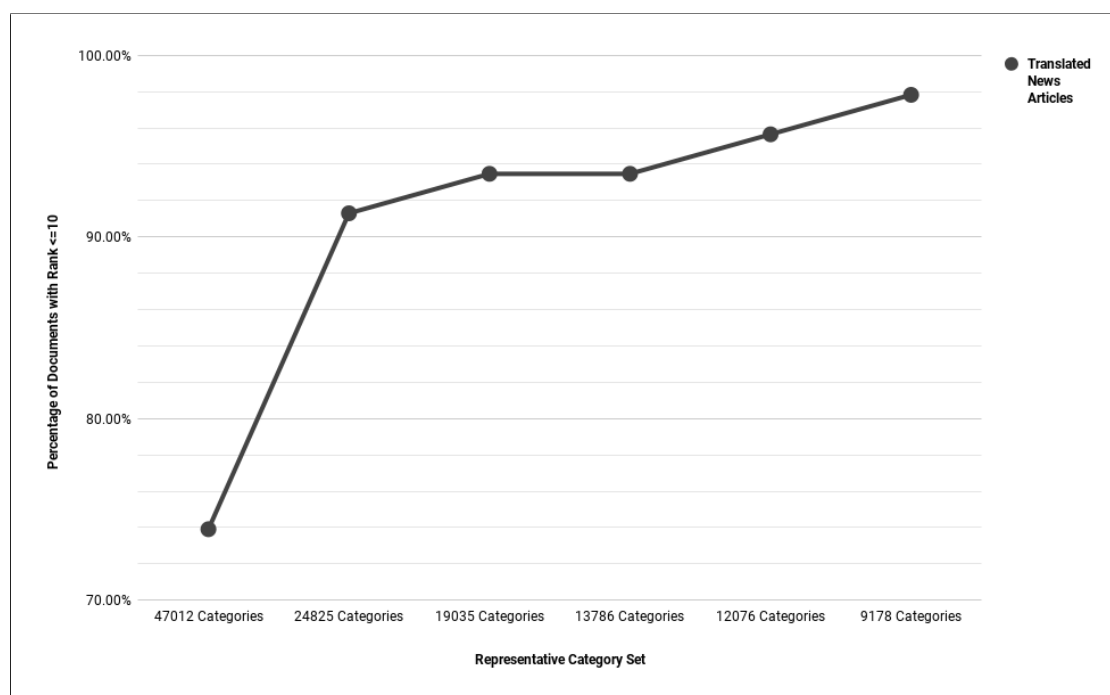


Figure 4.2: Percentage of Documents with Rank  $\leq 10$  to the Total number of Documents in the Collection for Translated News Articles



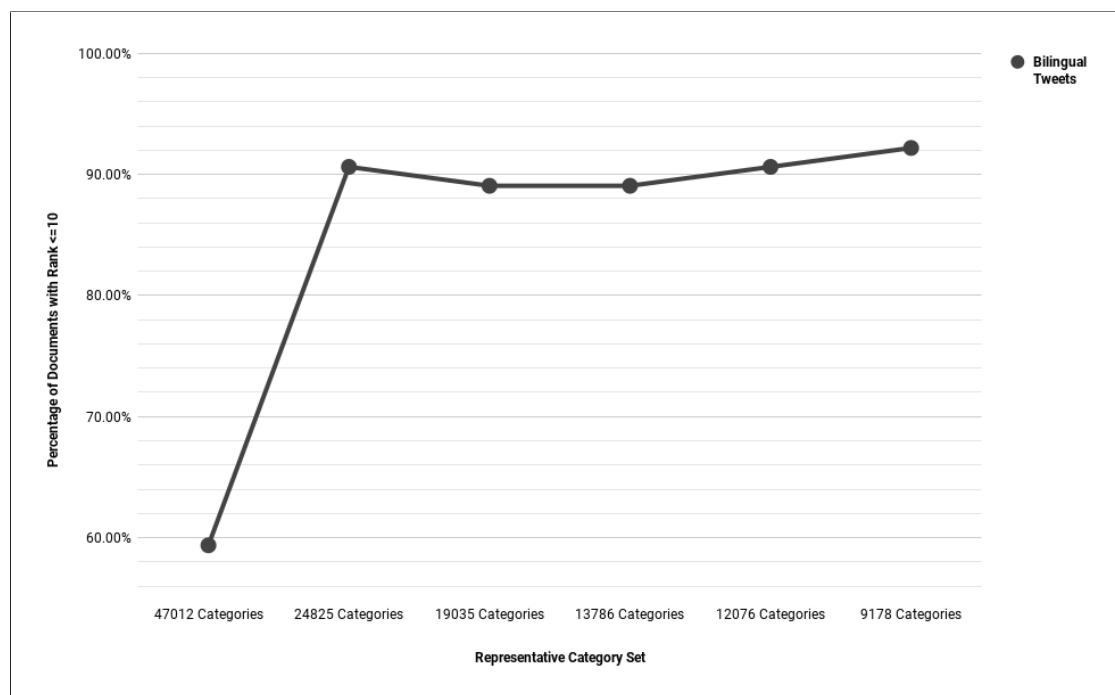


Figure 4.3: Percentage of Documents with Rank  $\leq 10$  to the Total number of Documents in the Collection for Bilingual Tweets

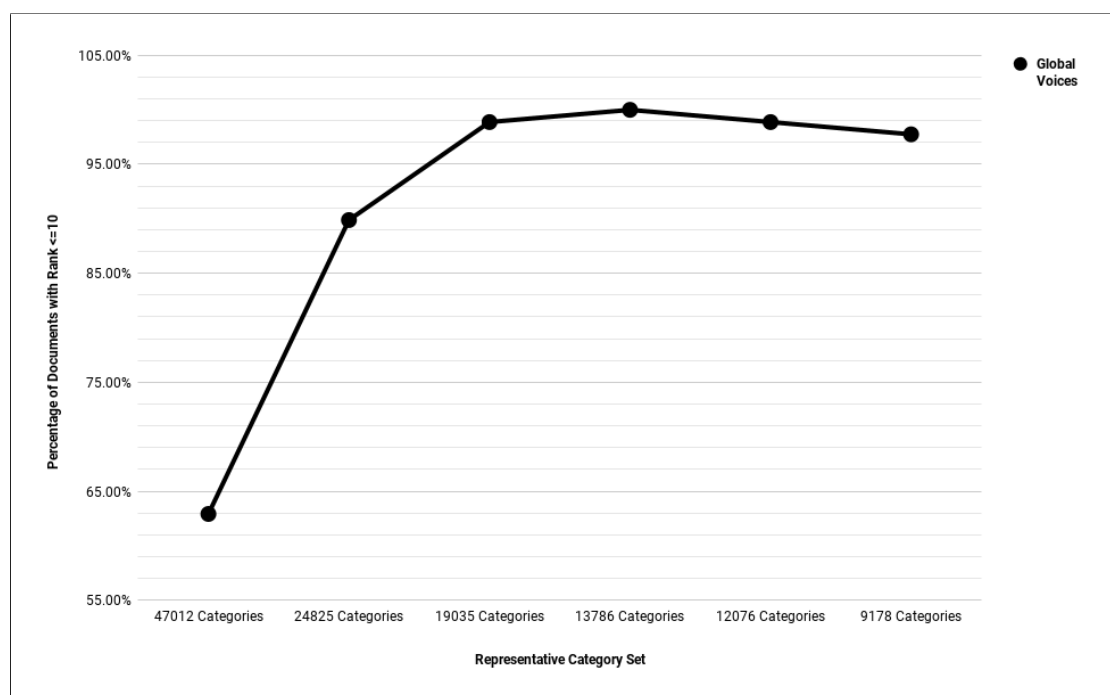


Figure 4.4: Percentage of Documents with Rank  $\leq 10$  to the Total number of Documents in the Collection for Global Voices

One can see that using 24825 Categories as representative category set gives best results for Wikipedia articles. However, for Birzeit News Story, Tweets and Global Voices test documents, 9178 Categories representative category set gives the best results.

Figures 4.5, 4.6, 4.7 and 4.8 show percentages of documents whose corresponding documents are found in the first 5 result documents in all similarity tests.

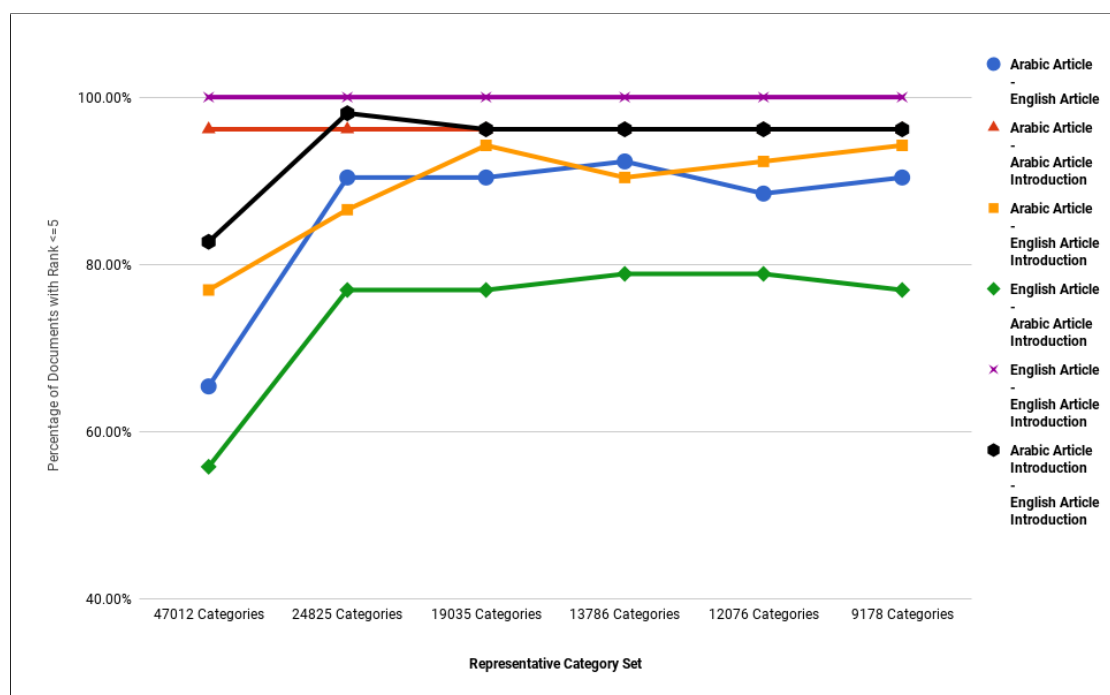


Figure 4.5: Percentage of Documents with Rank  $\leq 10$  to the Total number of Documents in the Collection for Wikipedia Featured Articles

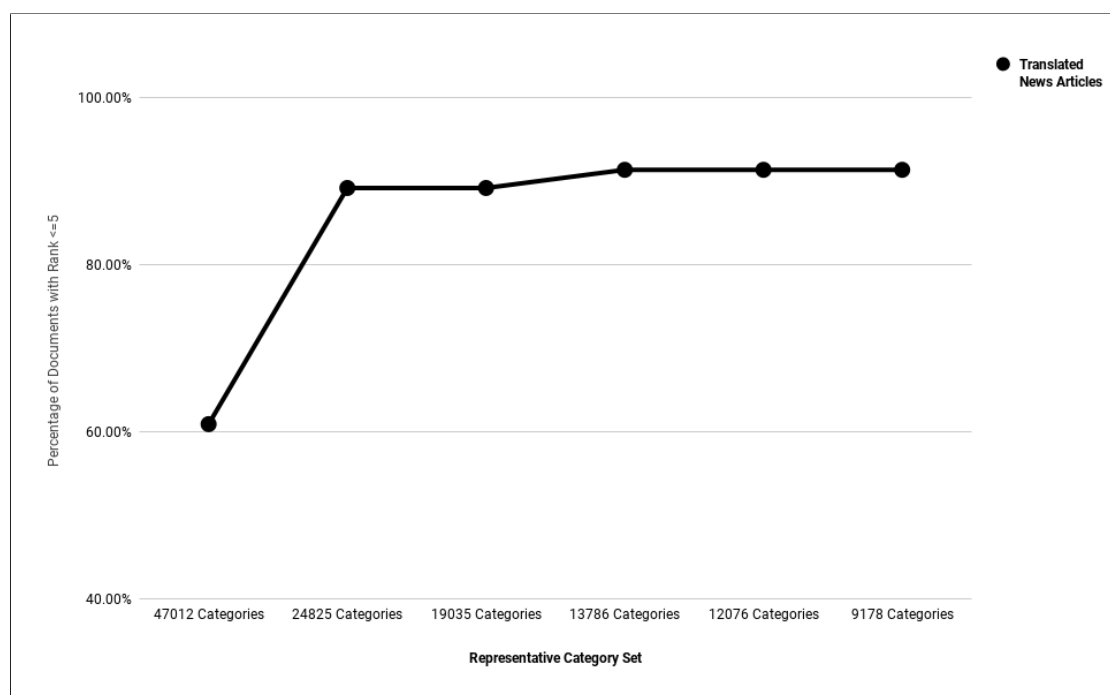


Figure 4.6: Percentage of Documents with Rank  $\leq 5$  to the Total number of Documents in the Collection for Translated News Articles

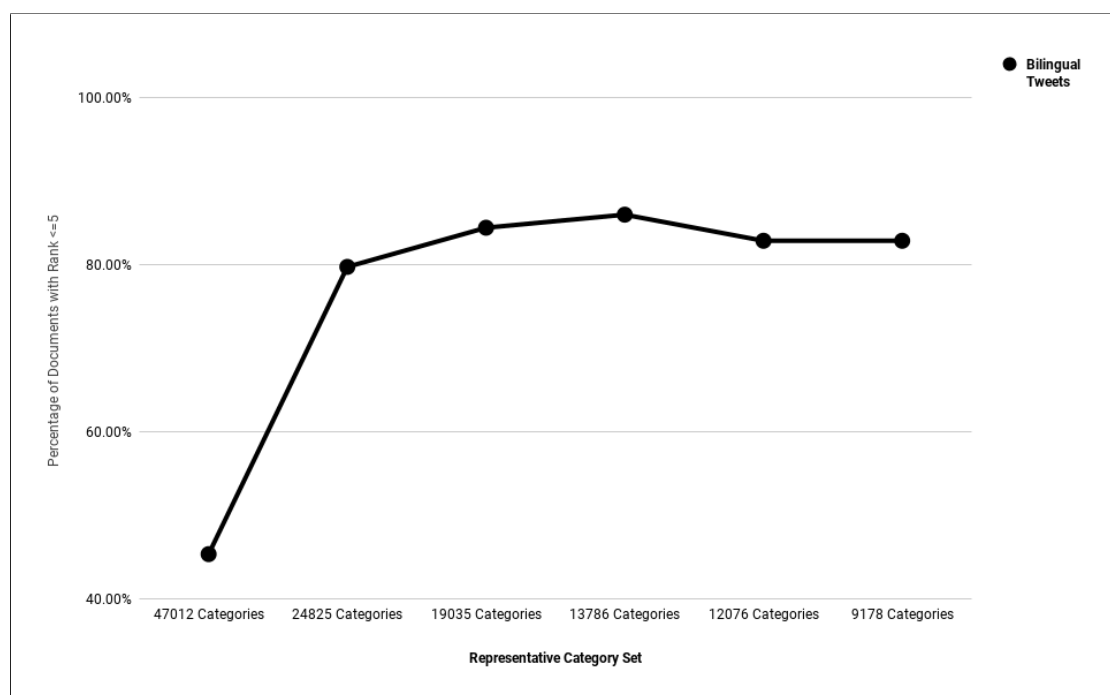


Figure 4.7: Percentage of Documents with Rank  $\leq 5$  to the Total number of Documents in the Collection for Bilingual Tweets

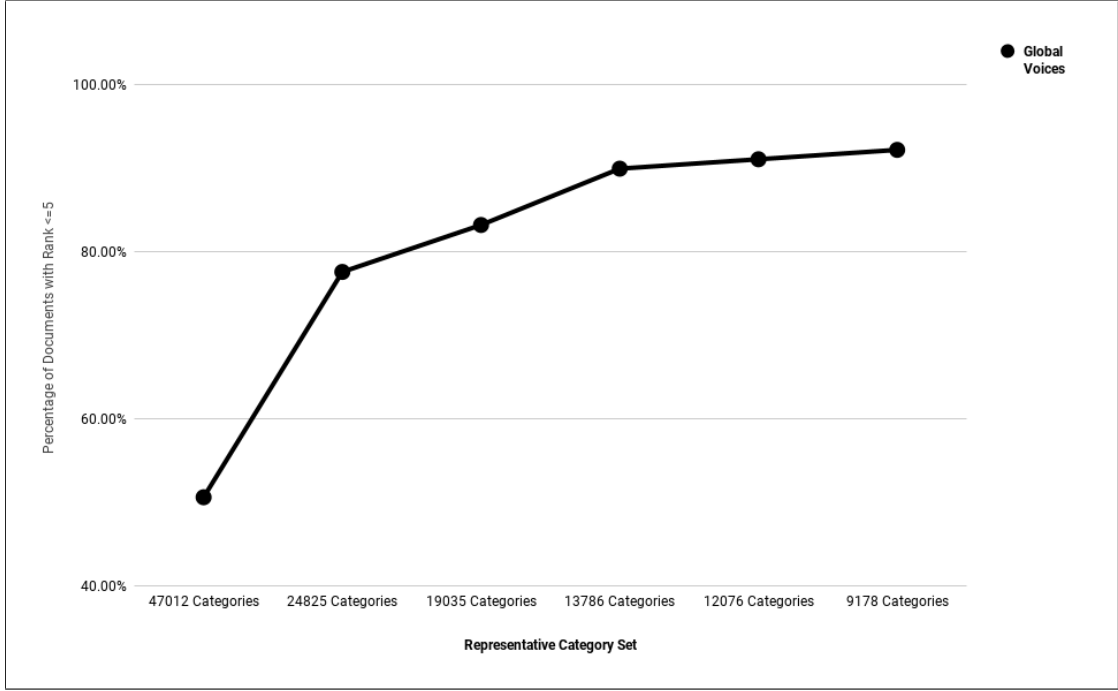


Figure 4.8: Percentage of Documents with Rank  $\leq 5$  to the Total number of Documents in the Collection for Global Voices

From these figures, one can see that no representative category set gives the best results in all tests. However, for Wikipedia Featured articles, using 13786 Categories as representative category set gives the best results in 4 tests.

For Birzeit News Stories, Tweets and Global Voices tests, using 13786 Categories or 9178 Categories as representative category set gives best results in 2 tests.

Figures 4.9, 4.10, 4.11 and 4.12 show percentages of documents whose corresponding documents are found as the first result documents in all similarity tests.

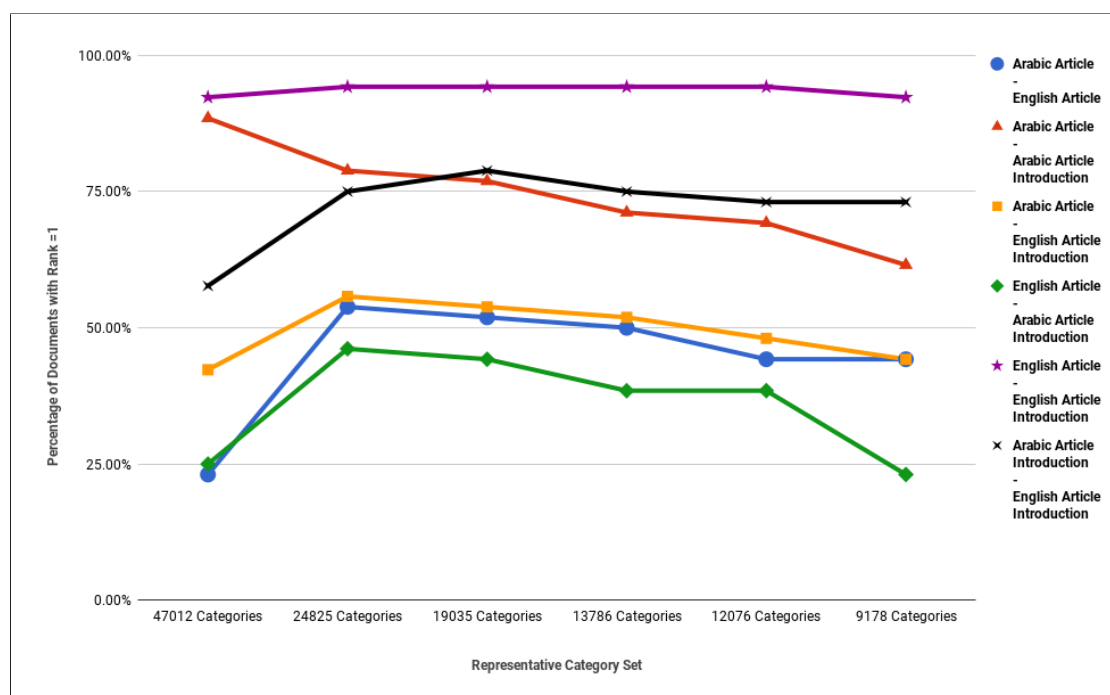


Figure 4.9: Percentage of Documents with Rank = 1 to the Total number of Documents in the Collection for Wikipedia Featured Articles

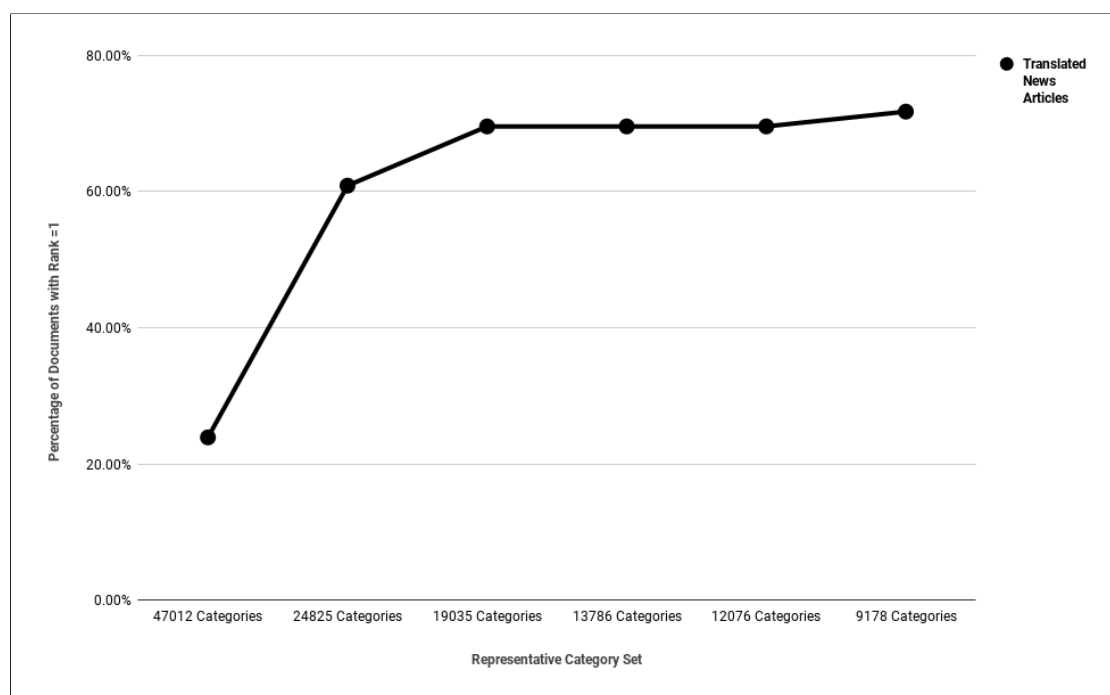


Figure 4.10: Percentage of Documents with Rank = 1 to the Total number of Documents in the Collection for Translated News Articles



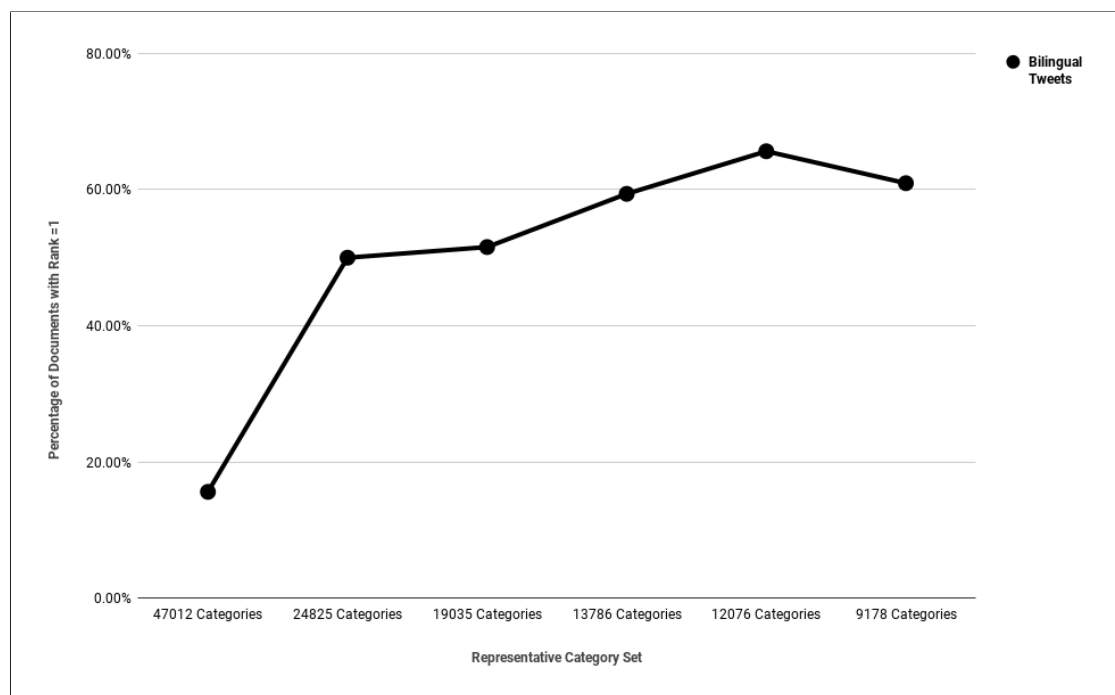


Figure 4.11: Percentage of Documents with Rank = 1 to the Total number of Documents in the Collection for Bilingual Tweets

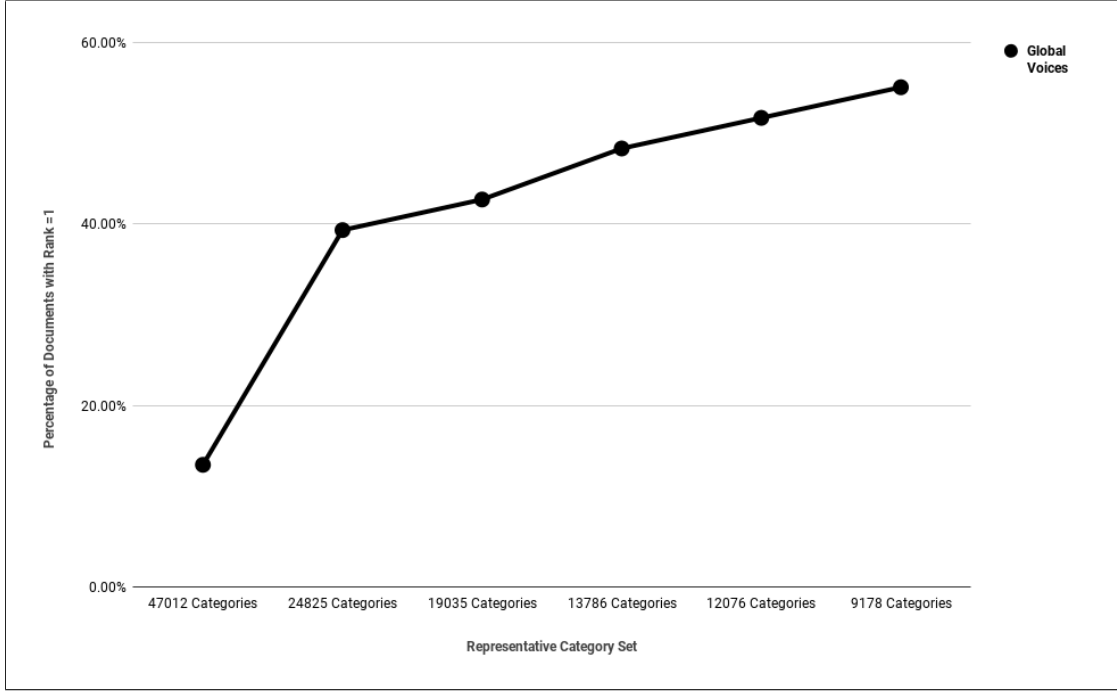


Figure 4.12: Percentage of Documents with Rank = 1 to the Total number of Documents in the Collection for Global Voices

For these figures, best results are obtained when using 24825 representative Categories in Wikipedia Mutual Articles tests. For Birzeit News stories, Tweets and Global Voices test, using 9178 representative Categories gives best results in most of the tests.

Referring to the similarity test results for Wikipedia featured articles, for long documents that don't have exact translated documents in the collection, using representative category set that has relatively big number of categories gives best similarity results. This might be related to the fact that noise in the category vector for such documents is minimized after accumulating the weights of Wikibase-Items. Wikipedia articles in Arabic are not actual translations of the English ones. They are related to each other because both talk about same concept.

On the other hand, using representative category set that has relatively small number of categories gives better results for short translated documents. Short documents have short category vectors (when dropping zero-weight components), so the weight of each Wikibase-Item starts to have more and more effect. In such cases, noisy Wikibase-Items might have big effect, so one needs to carefully select Wikibase-Items and minimize their numbers.

### 4.1.6 Using Part of Speech Tagging

The effect of removing some part of speech words on the results was tested. Similarity tests were executed using 24825 Categories as representative category two times; with and without applying POS Tagger. Table 4.7 shows the similarity test results for Arabic Articles as queries against collections of English Articles, Arabic Articles Introductions and English Articles Introductions with and without applying POS Tagger.

Table 4.7: Similarity Test Results for Queries of Arabic Articles and Documents of Different Collection with and without Applying POS Tagger

<b>Vector/Collect. Type</b>	<b>English Articles</b>	<b>Arabic Introductions</b>	<b>English Introductions</b>
24825 Categories without POS	<50,45,31>	<50,50,41>	<52,45,28>
24825 Categories with POS	<51,47,28>	<50,50,41>	<52,45,29 >
Improvement when using POS (%)	<2,4.4,9.7>	<0.0,0.0,0.0>	<0.0,0.0,3.6>

Table 4.8 shows the similarity test results for English Articles as queries against collections of Arabic Article Introductions and English Introductions with and without applying POS Tagger.

Table 4.8: Similarity Test Results for Queries of English Articles and Documents of Different Collection with and without Applying POS Tagger

<b>Vector/Collect. Type</b>	<b>Arabic Introductions</b>	<b>English Introductions</b>
24825 Categories without POS	<48,40,23>	<52,52,49>
24825 Categories with POS	<48,40,24>	<52,52,49>
Improvement when using POS (%)	<0.0,0.0,4.3>	<0.0,0.0,0.0>

Table 4.9 shows the similarity test results for Arabic Article Introduction as queries against a collection of English Article Introductions with and without applying POS Tagger.

Table 4.9: Similarity Test Results for Arabic Article Introductions and English Article Introductions with and without Applying POS Tagger

<b>Vector/Collect. Type</b>	<b>English Introductions</b>
24825 Categories without POS	<51,50,34>
24825 Categories with POS	<51,51,39>
Improvement when using POS (%)	<0.0,2.0,14.7>

Table 4.10 shows the similarity test results for Arabic Tweets as queries against a collection of English Tweets with and without applying POS Tagger.

Table 4.10: Similarity Test Results for Tweets with and without Applying POS Tagger

<b>Vector/Collect. Type</b>	<b>Result Vector</b>
24825 Categories without POS	<41,40,21>
24825 Categories with POS	<42,41,28>
Improvement when using POS (%)	<2.4 ,2.5,33.3>

Results are visualized in figures 4.13, 4.14 and 4.15

Figure 4.13 shows percentages of documents whose corresponding documents are found in the first 10 documents in the results list.

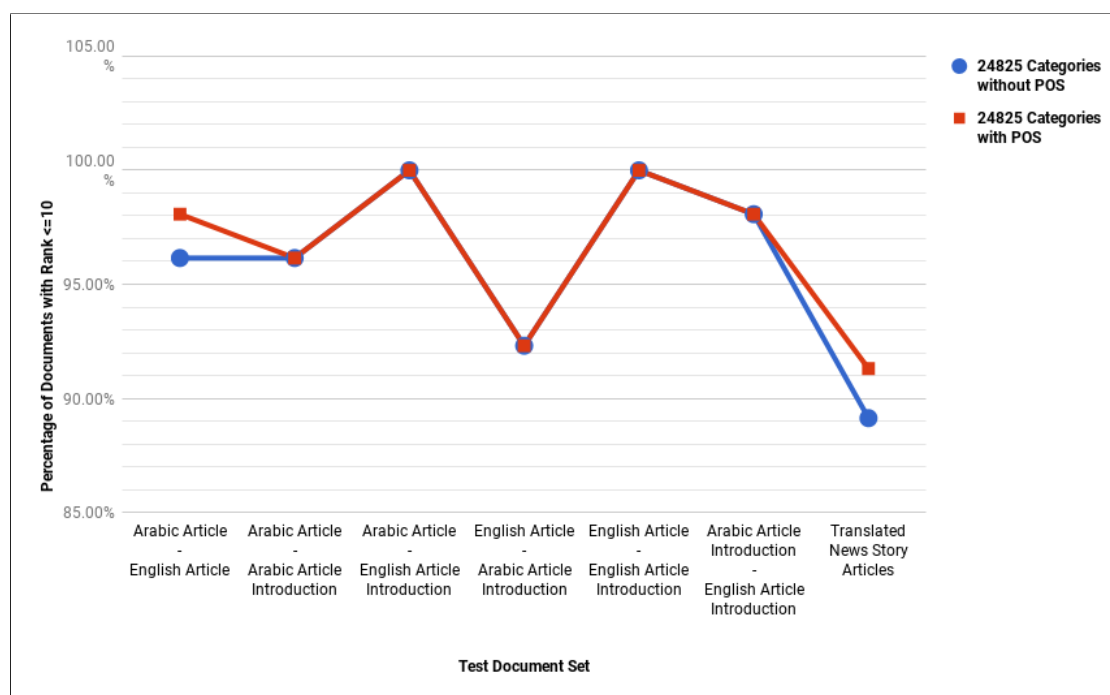


Figure 4.13: Percentage of Documents with Rank  $\leq 10$  to the Total number of Documents in the Collection with and without Applying POS Tagger

Figure 4.14 shows percentages of documents whose corresponding documents are found in the first 5 documents in the results list.

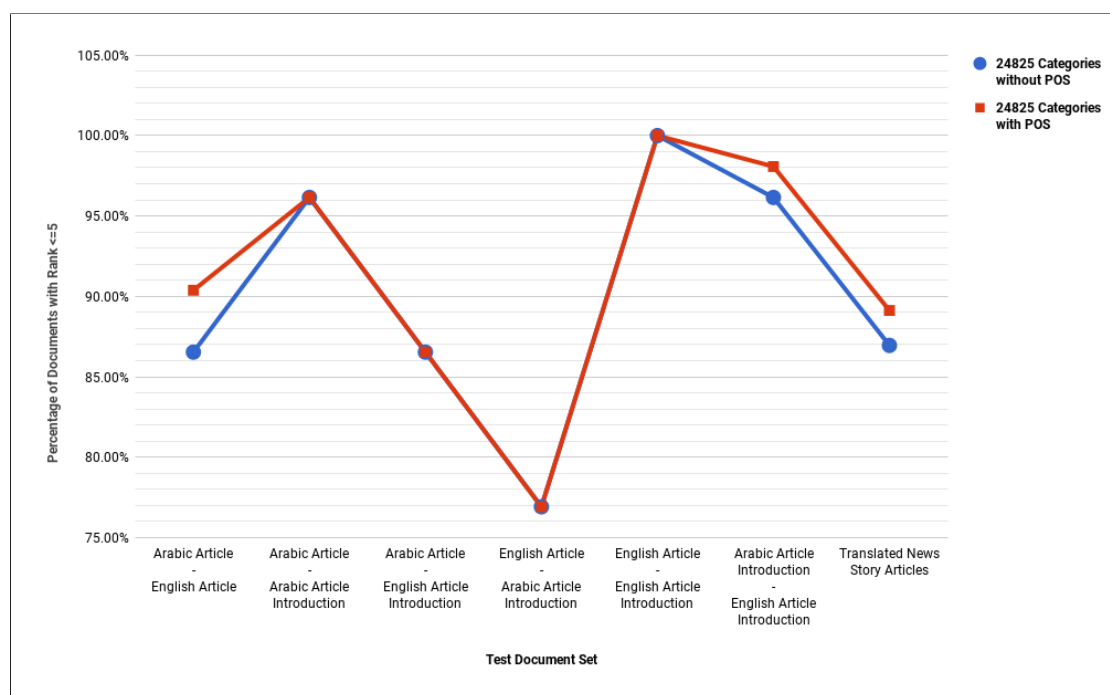


Figure 4.14: Percentage of Documents with Rank  $\leq 5$  to the Total number of Documents in the Collection with and without Applying POS Tagger

Figure 4.15 shows percentages of documents whose corresponding documents are found as first document in the results list.

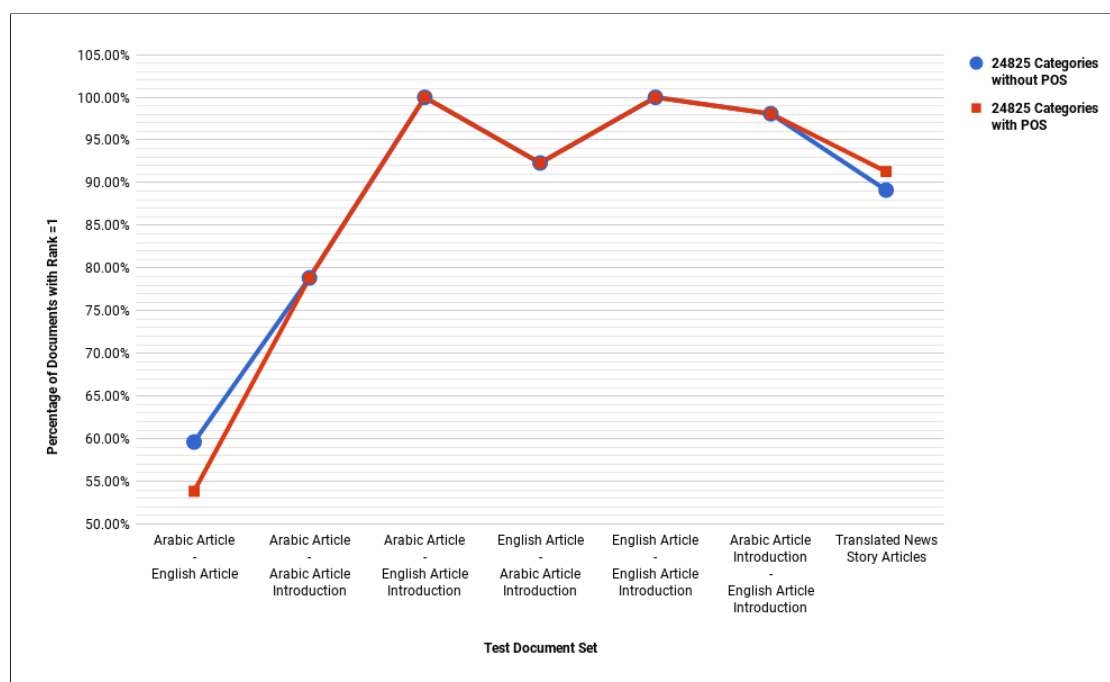


Figure 4.15: Percentage of Documents with Rank = 1 to the Total number of Documents in the Collection with and without Applying POS Tagger

Applying POS Tagger improves the results in most of the tests above.

### 4.1.7 Using Category Links

To improve the similarity tests results when using Category vectors to represents documents and queries, the concept of Category Links has been introduced. Each Wikipedia category has parent categories. These child-parent relations were retrieved and stored in the database at the time of parsing the dumps. When finding the similarity between each pair of the category vectors, only the weight of the categories that exist in both of the vectors are multiplied. For short documents, the category vector may not contain many of the representative categories.

In Wikipedia, two articles may share the same grand-parent categories, but not the parent ones. When finding the cosine similarity between a pair of category vectors, this relation is not taken into account, so the corresponding component weights will be producing zero.

To overcome that, the cosine similarity algorithm for the category vectors has been modified to include substituting the category with its parents in each of the vectors if this category doesn't exist in the other one (query and document

vectors).

If using category links in similarity tests improves the results, then this effect should clearly appear when using minimum number of representative categories to represent the vectors. To study this effect, similarity tests were executed using 9178 representative categories two times; with and without category links. Table 4.11 shows the similarity test results for Arabic Articles as queries against collections of English Articles, Arabic Articles Introductions and English Articles Introductions with and without Category Links.

Table 4.11: Similarity Test Results for Queries of Arabic Articles and Documents of Different Collection with and without Using Category Links

<b>Vector/Collect. Type</b>	<b>English Articles</b>	<b>Arabic Introductions</b>	<b>English Introductions</b>
9178 Categories without Category Links	<52,47,23>	<50,50,32>	<51,49,23>
9178 Categories with Category Links	<52,48,23>	<50,50,33>	<51,48,23>
Improvement when using Category Links (%)	<0.0,2.1,0.0>	<0.0,0.0,3.1>	<0.0,-2.1,0.0>

Table 4.12 shows the similarity test results for English Articles as queries against collections of Arabic Article Introductions and English Introductions with and without using category links.

Table 4.12: Similarity Test Results for Queries of English Articles and Documents of Different Collection with and without Using Category Links

<b>Vector/Collect. Type</b>	<b>Arabic Introductions</b>	<b>English Introductions</b>
9178 Categories without Category Links	<47,40,12>	<52,52,48>
9178 Categories with Category Links	<47,41,12>	<52,52,47>
Improvement when using Category Links (%)	<0.0,2.5,0.0>	<0.0,0.0,-2.1>



Table 4.13 shows the similarity test results for Arabic Article Introduction as queries against a collection of English Article Introductions with and without using Category Links.

Table 4.13: Similarity Test Results for Arabic Article Introductions and English Article Introductions with and without Using Category Links

<b>Vector/Collect. Type</b>	<b>English Introductions</b>
9178 Categories without Category Links	<50,50,38>
9178 Categories with Category Links	<50,50,40>
Improvement when using Category Links (%)	<0.0,0.0,5.3>

Table 4.14 shows the similarity test results for Arabic Tweets as queries against a collection of English Tweets with and without using category links.

Table 4.14: Similarity Test Results for Translated News Articles with and without Using Category Links

<b>Vector/Collect. Type</b>	<b>Result Vector</b>
9178 Categories without Category Links	<45,42,33>
9178 Categories with Category Links	<44,42,34>
Improvement when using Category Links(%)	<-2.2,0.0,3.0>

Table 4.15 shows the similarity test results for Arabic Tweets as queries against a collection of English Tweets with and without using category links.

Table 4.15: Similarity Test Results for Tweets with and without Using Category Links

<b>Vector/Collect. Type</b>	<b>Result Vector</b>
9178 Categories without Category Links	<59,53,39>
9178 Categories with Category Links	<59,52,40>
Improvement when using Category Links(%)	<0.0,-1.9,2.6>

Table 4.15 shows the similarity test results for Global Voices parallel corpus with and without using category links.

Table 4.16: Similarity Test Results of Global Voices Parallel Corpus with and without Using Category Links

Vector/Collect. Type	Result Vector
9178 Categories without Category Links	<87, 82,49>
9178 Categories with Category Links	<86,80,48>
Improvement when using Category Links(%)	<-1.1,-2.4,-2.0>

To get a better picture for the effect of using category links on the results, percentages of documents whose corresponding documents are found in the first 10, 5 and 1 result documents are visualized for both cases: with and without using category links.

Figure 4.16 shows percentages of documents whose corresponding documents are found in the first 10 result documents in all similarity tests.

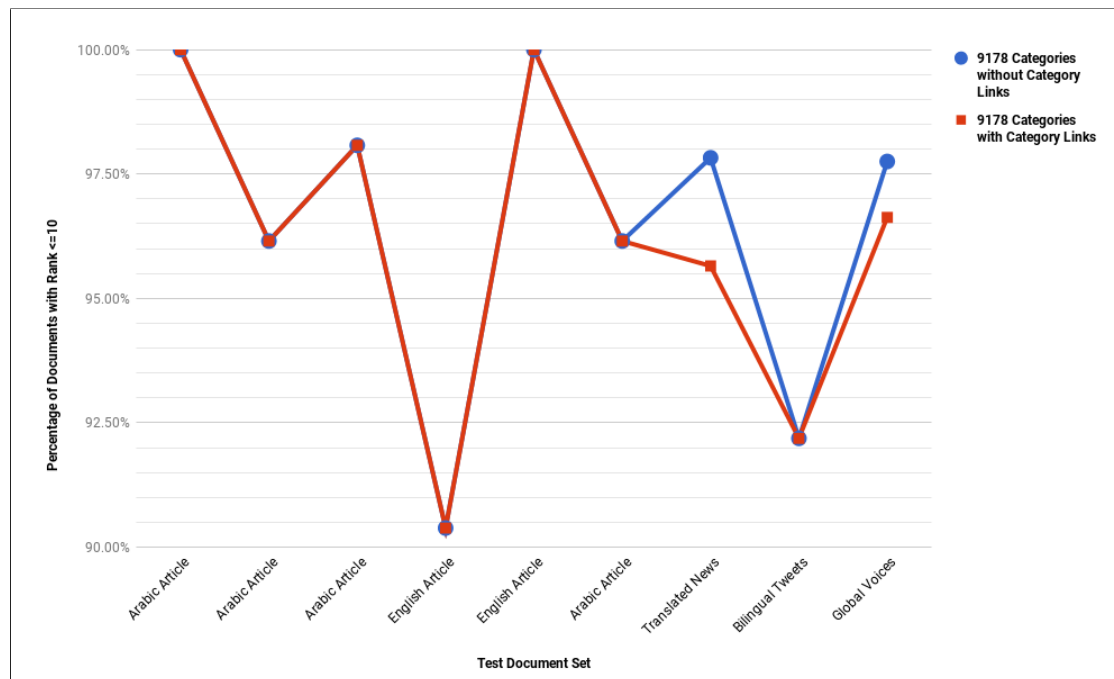


Figure 4.16: Percentage of Documents with Rank  $\leq 10$  to the Total number of Documents in the Collection

Figure 4.17 shows percentages of documents whose corresponding documents are found in the first 5 result documents in all similarity tests.

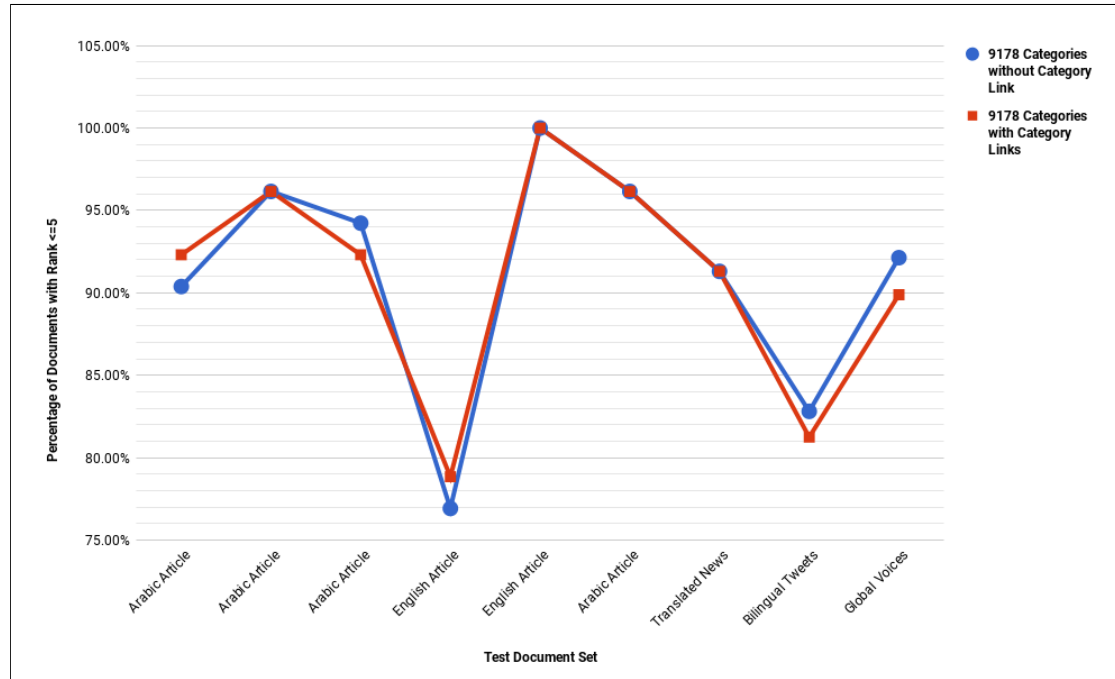


Figure 4.17: Percentage of Documents with Rank  $\leq 5$  to the Total number of Documents in the Collection

Figure 4.18 shows percentages of documents whose corresponding documents are found as the first result documents in all similarity tests.

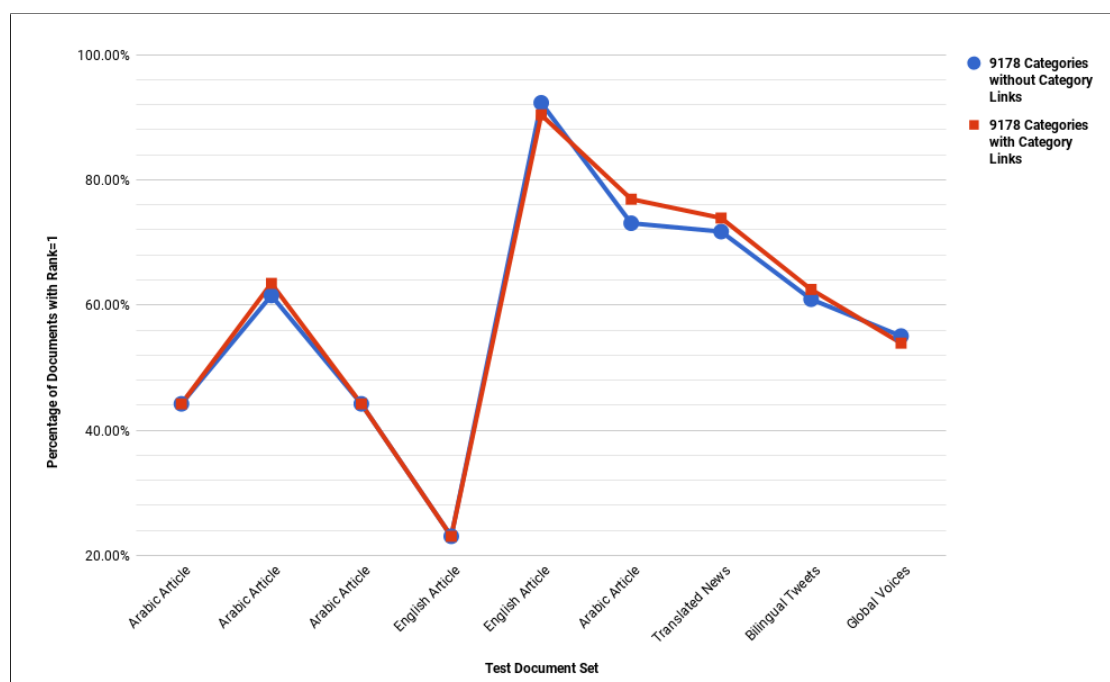


Figure 4.18: Percentage of Documents with Rank Equals to 1 to the Total number of Documents in the Collection

The results figures above show that using category links in the similarity calculations doesn't significantly affect the results. To explain this, the word-category distribution has been reviewed and found to be dense. Which means, each word exists in a large number of categories, so each category vector contain most of the representative categories. Another observation is that the category-weight distributions is wide and sparse; some categories have very high weights and other have very low weights. This issue will be discussed in the next section.

### 4.1.8 Vector Length

Noisy components in the similarity vectors may drop the similarity accuracy down, even if their weights are relatively small. One way to eliminate the noise is to take only the  $n$  components with highest weight in similarity calculations, and skip the rest. For each document vector, this can be done in two steps:

- Sort the vector components descending by weight
- Take the top  $n$  components

To evaluate the effect of dropping weak components of the Category vectors on similarity results, similarity tests were executed for  $n = 1000, 500, 100, 50, 30$  and 10 using each of the representative category sets.

To have a better picture of the results, the similarity score defined in 3.16 is used, and to evaluate the scores of each n-vector test among all the similarity tests, the scores of each test are sorted in descending order and ranked. The score for each n-vector test equals the sum of all the ranks. The best n-vector is the one with the lowest total rank.

Similarity tests have been executed for the following query-collection pairs:

- Arabic Wikipedia Featured Article, English Wikipedia Featured Articles
- Arabic Wikipedia Featured Article, Arabic Wikipedia Featured Article Introductions
- Arabic Wikipedia Featured Article, English Wikipedia Featured Article Introductions
- English Wikipedia Featured Article, Arabic Wikipedia Featured Article Introductions
- English Wikipedia Featured Article, English Wikipedia Featured Article Introductions
- Arabic Wikipedia Featured Article Introduction, English Wikipedia Featured Article Introductions
- Arabic Tweet, English Tweets
- Arabic News Article, English News Articles
- Arabic Article in Global Voices Parallel Corpus, English Articles in Global Voices Parallel Corpus

And each of the tests above is executed for vectors generated from:

- 47012 Categories
- 24825 Categories
- 19031 Categories
- 13786 Categories
- 12076 Categories

- 9178 Categories

Tests are also executed when trimming the vector to the following number of non-zero components:

- All components
- 1000 Components
- 500 Components
- 100 Components
- 50 Components
- 30 Components
- 10 Components

So, a total of 378 tests were executed to study the effect of the vector length on the results. Figures 4.19, 4.20, 4.21, 4.22, 4.23 and 4.24 show the scores obtained for the different representative category sets.

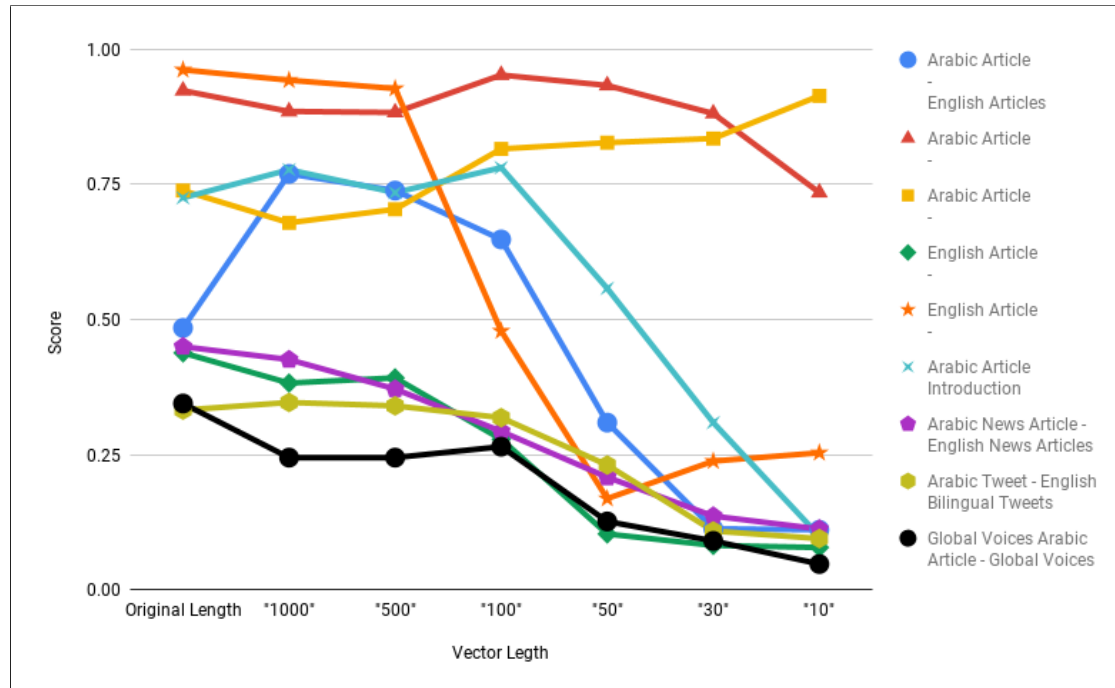


Figure 4.19: Similarity Test Scores when Using 47012 Categories

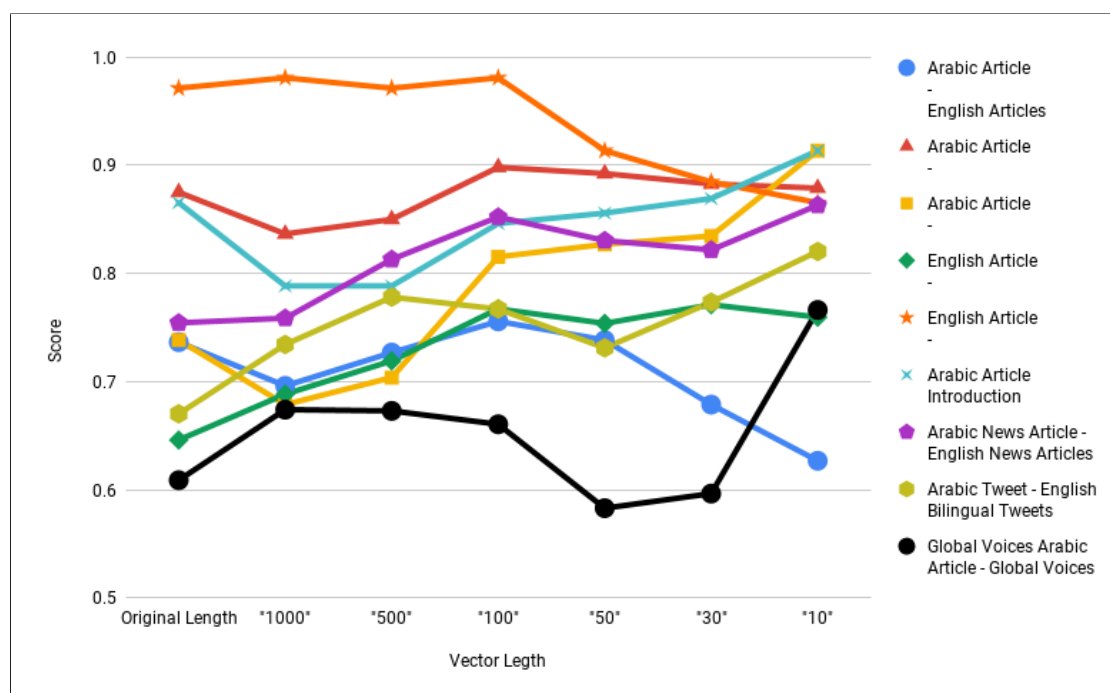


Figure 4.20: Similarity Test Scores when Using 24825 Categories

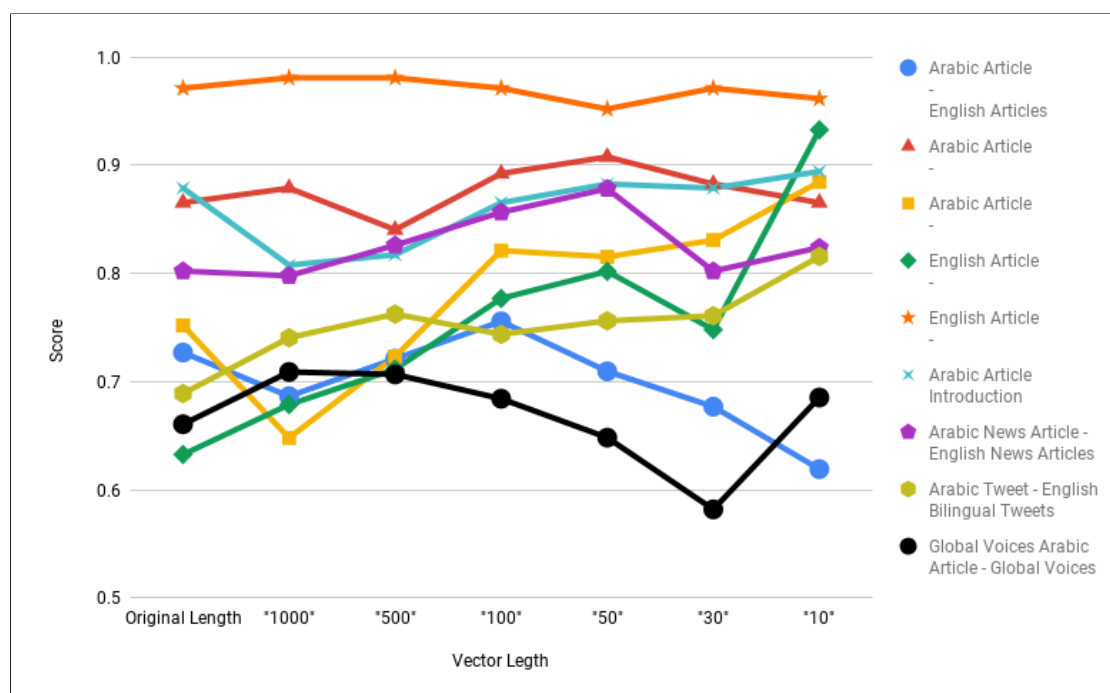


Figure 4.21: Similarity Test Scores when Using 19031 Categories



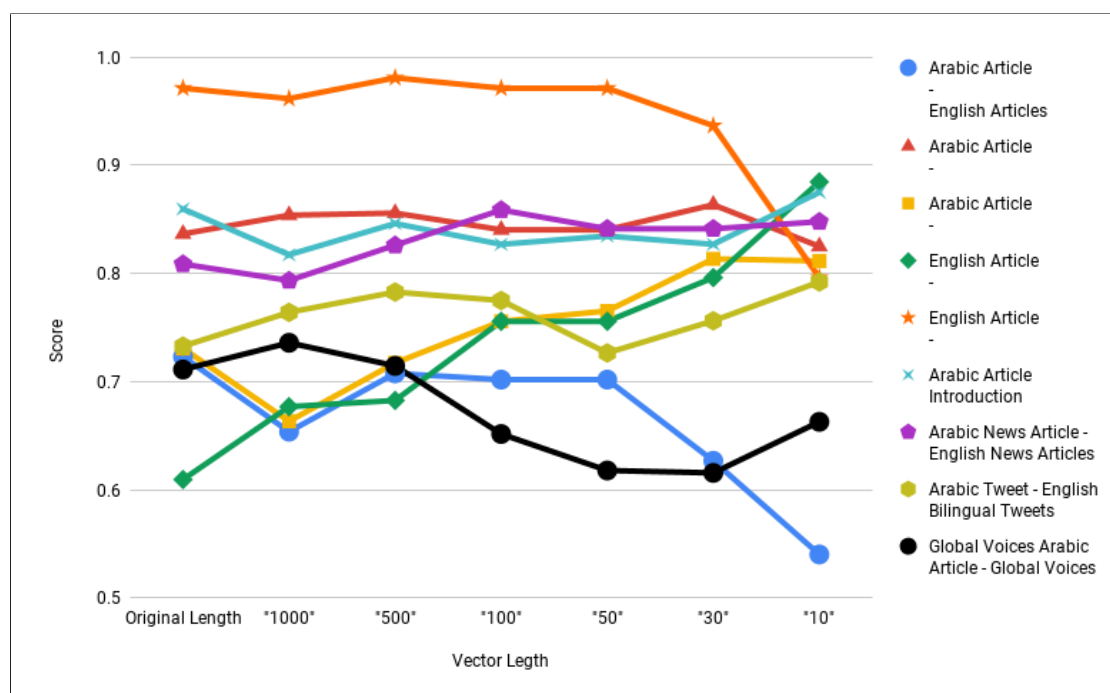


Figure 4.22: Similarity Test Scores when Using 13786 Categories

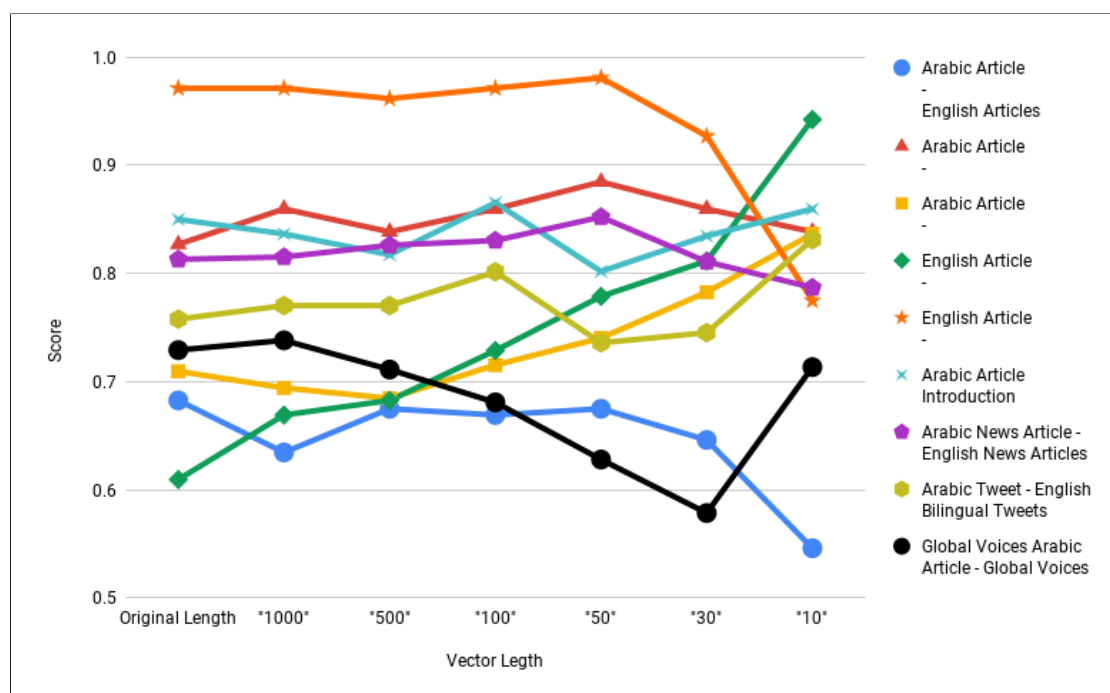


Figure 4.23: Similarity Test Scores when Using 12076 Categories

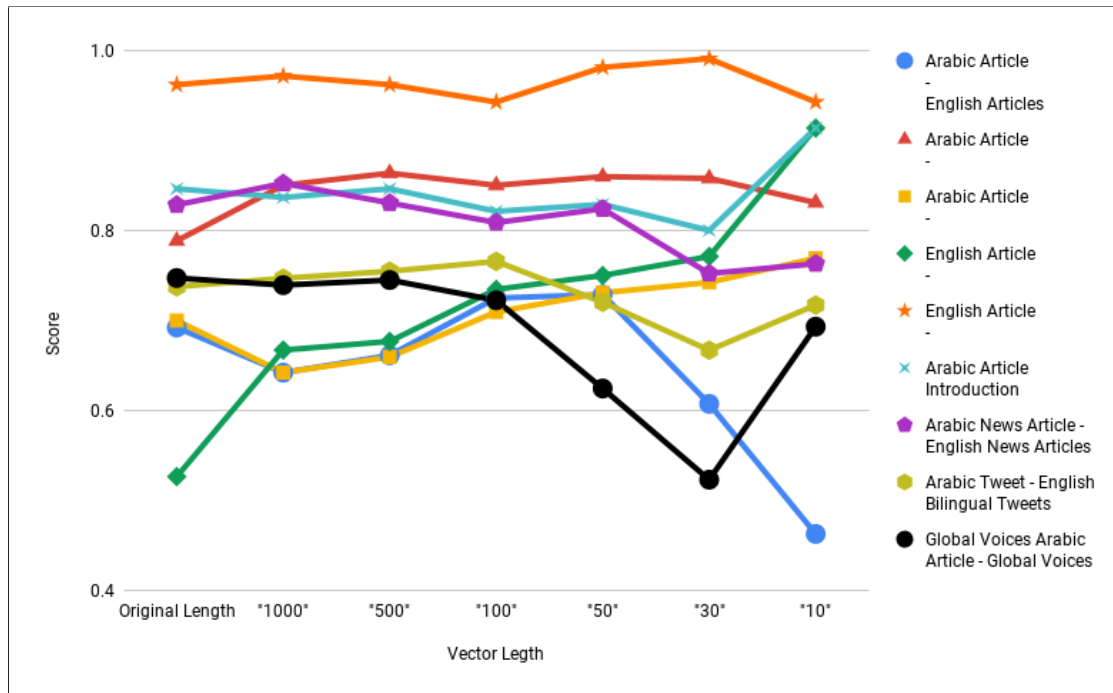


Figure 4.24: Similarity Test Scores when Using 9178 Categories

The aggregate rank score is calculated for each test as follows:

- Group all 378 tests by their query-collections pairs, which produces the 9 pairs listed above
- Sort the score for each pair group in descending order then add a rank for them. At this point all the 378 tests are ranked
- Group the tests by representative category and vector length, and find the aggregate rank for each group by adding the tests ranks

To explain the tests and the result analysis, figure 4.25 shows a snapshot of the actual calculations.

	A	B	M	N	O	P	Q	R	S	T	U	V	W
1					Translated News Articles (48)		Bilingual Tweets (64)		Global Voices (89)		Total (Sum)	Total (Rank)	Rank (Total(Rank)) - ASC
2	Vector Type	Vector Length	Arabic Article Introduction - English Article Introductions		Arabic News Article - English News Articles		Arabic Tweet - English Bilingual Tweets		Global Voices Arabic Article - Global Voices English Article		Total Score	Total (Rank)	
3		Original Length	0.725000	39	0.450000	36	0.332813	38	0.344944	36	5.295064	286	39
4		"1000"	0.776923	37	0.426087	37	0.346875	36	0.244944	38	5.562137	235	36
5		"500"	0.734615	38	0.371739	38	0.340625	37	0.244944	39	5.420769	248	37
6	47012 Categories	"100"	0.780769	36	0.293478	39	0.318750	39	0.265169	37	4.779320	275	38
7		"50"	0.557692	40	0.208696	40	0.231250	40	0.126966	40	3.197681	324	40
8		"30"	0.309615	41	0.136957	41	0.109375	41	0.091011	41	2.239266	341	41
9		"10"	0.100000	42	0.113043	42	0.095313	42	0.048315	42	1.658594	376	42
10		Original Length	0.865385	9	0.754348	34	0.670313	34	0.608989	30	6.866341	191	22
11		"1000"	0.788462	34	0.758696	33	0.734375	27	0.674157	19	6.836459	225	34
12		"500"	0.788462	35	0.813043	22	0.778125	7	0.673034	20	7.023818	181	17
13	24825 Categories	"100"	0.846154	16	0.852174	5	0.767188	12	0.660674	22	7.343497	85	1
14		"50"	0.855769	14	0.830435	11	0.731250	29	0.583146	32	7.125600	153	8
15		"30"	0.869231	8	0.821739	20	0.773438	9	0.596629	31	7.112960	149	7

Figure 4.25: A Snapshot of the Actual Score Analysis

The first step is to execute all the similarity tests then find the score for each of them using equation 3.16. The score values are in columns M, O, Q, R and U. The second step is grouping tests by query-collection pair. In this case. we have the 9 groups mentioned above. Then for each group, scores are sorted in descending order then ranked. So the highest score in each group has rank 1, the second score has rank 2, etc. These ranks are stored in columns N, P, R, T and V. The third step is to group tests by representative category set-vector length pairs. ie. We should have 42 groups (6 representative category set groups and 7 vector length groups). Then, ranks (Columns N, P, R, T and V) in each groups are aggregated, in column V. The last step is to sort the aggregated ranks in column V in descending order the rank them in column W. The best representative category - vector length group is the one with rank 1. Figure 4.26 shows the aggregate scores.

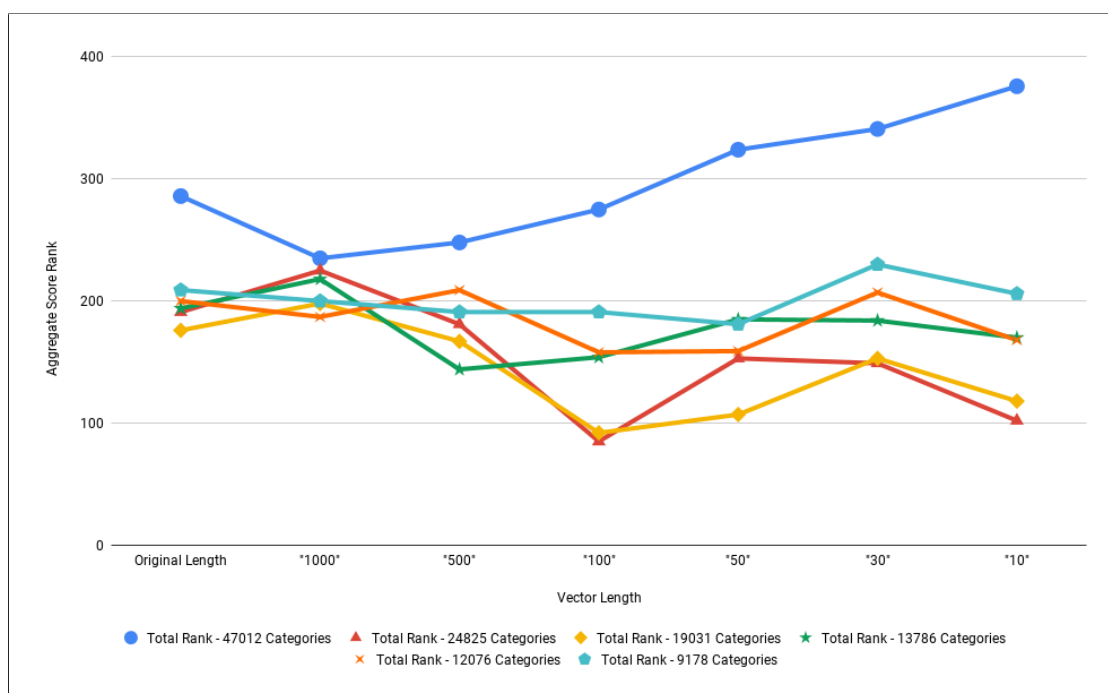


Figure 4.26: Similarity Test Aggregate Score

The figure above shows that the best representative category set-vector length pairs that give the best ranks in most of the groups are 24825 Categories - 100 (aggregate score = 81), 19031 Categories - 100 (aggregate score = 88) and 24825 Categories - 10 (aggregate score = 90), respectively. One would choose 24825 Categories - 100 as the best pair to use in similarity tests, since it has the lowest total rank.

We reviewed the sub-scores for each of these three pairs to check if the aggregate score for one of them is dropped by some tests. We found that that using 24825 Categories - 10 gives the best results for short and mid length documents (aggregate score = 6). Whereas, for long documents, 24825 Categories - 100 is still the best (score = 26).

Results above show performance improvement in computing similarity between documents of different lengths and in same or different languages, when using Wikipedia categories to represent these documents as vectors, compared to using Articles. Such approach in computing similarity may be used in search engines, which rely on finding similarity between query and collection documents. Plagiarism detection is another interesting area that may benefit from such system in finding the relatedness between articles.

The difference between number of articles and categories used to representing

the documents is another improvement to consider as well. Similarity performance is the most important feature to consider, however the efficiency of representing documents using the concepts and computing the cosine similarity between vectors is also a key feature, specially for real time systems. Having a short list of concepts that can truly represent the documents means minimize the time and resources consumed by the system to retrieve the results.

## 4.2 YAGO Facts Translation

We ran the algorithm proposed to translate YAGO facts on YAGO sample facts file, which contains 18845 facts for 33 relations. Then we analyzed the results in two directions: the translation status (translated, semi translated or not translated), translation quality by human ranking and compared to a context search engine.

### 4.2.1 Fact Translation Status

We evaluated the translation status for facts assigning each of them one of the following status:

- **Translated** if the fact is completely translated. This includes correct translation, transliteration and incorrect translation
- **Semi Translated** if fact is not translated and Arabic description is appended to it, or if the major part of the fact is not translated
- **Not Translated** when no Wikipedia title, Wikidata label, Wikidata alias or Wikidata description in Arabic is found for the English title

Table 4.17 shows the percentage of subjects with each status.

Since Google Translate API is used to translate those objects couldn't be translated using Wikipedia and Wikidata, no object can be marked as not translated. We evaluated only objects that have been translated using Wikipedia and Wikidata. Table 4.18 shows the percentage of objects that have been translated using Wiki along with percentage of completely and translated objects in them.

Table 4.17: Translated YAGO Fact Subject Translation Status per Property

<b>Relation</b>	<b>Complete Translation (%)</b>	<b>Semi Translation (%)</b>	<b>No Translation (%)</b>
hasGender	97.33	2.67	0.00
actedIn	99.92	0.08	0.00
created	3.26	0.00	0.00
dealsWith	100.00	0.00	0.00
diedIn	100.00	0.00	0.00
directed	98.61	0.00	1.39
exports	100.00	0.00	0.00
graduatedFrom	100.00	0.00	0.00
happenedIn	100.00	0.00	0.00
hasAcademicAdvisor	100.00	0.00	0.00
hasCapital	100.00	0.00	0.00
hasChild	97.88	0.00	2.12
hasCurrency	100.00	0.00	0.00
hasMusicalRole	100.00	0.00	0.00
hasNeighbor	100.00	0.00	0.00
hasOfficialLanguage	100.00	0.00	0.00
hasWebsite	100.00	0.00	0.00
hasWonPrize	99.30	0.00	0.70
imports	100.00	0.00	0.00
influences	100.00	0.00	0.00
isAffiliatedTo	100.00	0.00	0.00
isCitizenOf	98.96	1.04	0.00
isKnownFor	100.00	0.00	0.00
isLeaderOf	100.00	0.00	0.00
isLocatedIn	98.66	1.34	0.00
isMarriedTo	100.00	0.00	0.00
isPoliticianOf	100.00	0.00	0.00
livesIn	100.00	0.00	0.00
owns	100.00	0.00	0.00
participatedIn	1.33	0.00	0.00
playsFor	100.00	0.00	0.00
wasBornIn	98.24	1.76	0.00
wroteMusicFor	100.00	0.00	0.00

Table 4.18: Translated YAGO Facts Objects Translation Status per Property

<b>Relation</b>	<b>Translated using Wiki (%)</b>	<b>Complete Translation (%)</b>	<b>Semi Translation (%)</b>
hasGender	100.00	100.00	0.00
actedIn	84.02	54.85	44.74
created	25.50	0.20	1.89
dealsWith	99.12	100.00	0.00
diedIn	98.18	94.44	5.56
directed	88.89	53.13	46.88
exports	96.85	99.19	0.81
graduatedFrom	78.95	90.00	10.00
happenedIn	100.00	100.00	0.00
hasAcademicAdvisor	100.00	100.00	0.00
hasCapital	94.12	96.88	0.00
hasChild	74.60	90.78	8.51
hasCurrency	83.33	100.00	0.00
hasMusicalRole	50.00	100.00	0.00
hasNeighbor	97.03	100.00	0.00
hasOfficialLanguage	87.50	100.00	0.00
hasWebsite	100.00	100.00	0.00
hasWonPrize	78.25	100.00	0.00
imports	93.83	98.68	1.32
influences	79.34	67.40	32.60
isAffiliatedTo	88.79	100.00	0.00
isCitizenOf	100.00	100.00	0.00
isKnownFor	57.89	100.00	0.00
isLeaderOf	42.80	86.54	10.58
isLocatedIn	92.28	98.91	0.73
isMarriedTo	80.00	92.95	7.05
isPoliticianOf	100.00	100.00	0.00
livesIn	96.83	96.72	3.28
owns	49.74	42.59	56.58
participatedIn	62.33	1.22	0.23
playsFor	81.13	100.00	0.00
wasBornIn	92.94	91.77	8.23
wroteMusicFor	60.00	25.93	74.07



### 4.2.2 Fact Translation Quality: Human Ranking

To qualify the quality of the fact translation, we asked a group of 10 native Arabic speakers to manually judge the correctness of the translated facts. Each participant was presented 200 random English facts with their Arabic translation, and asked to give each subject and object a rank, as follows:

- -1: Translation is wrong
- 0: Subject or Object is not translated
- 1: Translation is weak, and hardly understandable
- 2: Translation is not that good, but acceptable
- 3: Translation is good

Table 4.19 summarizes the results.

Table 4.19: Translated YAGO Facts Human Ranking Summarization

	<b>Translated Subjects</b>	<b>Translated Objects</b>
Average Rank	02.68	02.51
Rank= -1 Percentage	04.26%	02.76%
Rank= 0 Percentage	01.25%	02.09%
Rank= 1 Percentage	02.34%	07.69%
Rank= 2 Percentage	06.43%	16.14%
Rank= 3 Percentage	85.71%	71.32%

### 4.2.3 Fact Translation Quality: Compare to Context Search Engine Results

Facts translation results were also compared to the results obtained when querying subjects and objects in an English-Arabic context search engine<sup>1</sup>. Context based search Subjects are used as queries in the search engine, and the translated subject is compared in the Arabic results, if it exists in at least 50% of the retrieved results, then it was marked as correct translation, otherwise it was marked as incorrect. If subject was not found in the retrieved results, it was marked as not found. In this way, we can measure how the the translated subject

---

<sup>1</sup><https://context.reverso.net/translation/>

is widely used as translation for the subject. Object sometimes was appended to the subject query to improve the accuracy of the retrieved results. The same procedure was repeated for the objects.

For example, the fact <United Kingdom><participatedIn><Operation Spring>is translated to <عملية ينبوع><شارك في><المملكة المتحدة>. When looking for Operation Spring in the context search engine, all the retrieved Arabic texts point to عملية سلة الربيع and عملية الربيع. Both ينبوع and سلة in Arabic are corresponding to Spring word in English, but the context that is linked to Spring Operation in English is pointing to الربيع in Arabic. So the Wiki translation here is marked as incorrect.

The fact <Prince (musician)><created><Little Red Corvette>is translated to <ليتل ريد كورفيت><صنع><برنس>. When looking for the object Little Red Corvette in the context search engine, retrieved Arabic texts contain mainly two phrase: ليتل ريد كورفيت, which is the Arabic transliteration, and الكورفيت الحمراء الصغيرة, which is the exact Arabic translation. So, the transliteration for this fact is considered as accepted translation.

Tests were executed for the same 200 random facts used in the previous sections. Results are summarized in table below

Table 4.20: Translated YAGO Facts Compared to Context Search Engine Results

	Subjects (%)	Objects (%)
Not Found	2.50	13.0
Incorrect Translation	0.50	13.5
Correct Translation	97.0	73.5

The results evaluated in the three sections above show that using Wikipedia and Wikidata (and lately Google Translate API) to translate such structured data to Arabic is efficient. Subject translation results are quite good and better than the results when translating objects. Subjects are usually well-known named-entities and used widely, whereas object are usually less known. Having such good results for the sample fact file may open the door to translate the whole YAGO knowledge base to Arabic, which will improve the Arabic quality Web content by adding more correlated terms to the total web content. It also may make is possible to make use of the foreign information retrieval systems that have been built over YAGO, such as question answering system [60] and domain-specific knowledge bases like event knowledge base [61]. It also might be usefully in generating unstructured data, like Wikipedia articles.

## Chapter 5

# Conclusion and Future Work

In this work, we proposed an approach to compute semantic relatedness between natural language texts, possibly in different languages. This new approach relies on using Wikipedia Categories to represent the query and documents in same or different languages, then using cosine similarity to find similarity values between vectors. The effect of selecting the category set to represent the documents, using POS taggers in document preprocessing, involving category links in similarity calculation and trimming vector to specific lengths are all studied. The empirical results showed that using category link and POS tagger had little improvement on the similarity results. Whereas selecting representative category set and vector length significantly affects the results. Our results showed also improvements in computing semantic relatedness compared to using Wikipedia articles (in ESA) in representing queries and documents, and the obtained accuracy was between 0.66 and 0.98, when using the best obtained representative category set - vector length pair along with applying POS tagger.

We also proposed a novel approach to translate structured data in YAGO knowledge base to Arabic. This approach relies on the Wikipedia and Wikidata items to translate facts subjects and objects. Google Translate API was used in translating the objects that couldn't be translated using Wikidata. Results were evaluated by human ranking and by comparing the translated results to the retrieved ones when querying these English subjects and objects in English-Arabic context search engine. Both evaluation methods showed high quality in translating both subjects and objects.

For future work in cross-lingual similarity, we may need to improve parsing the Wikipedia pages by applying more preprocessing stages. Storing the words as n-grams and storing words positions in postings may be another two things to implement, which may improve the similarity results for short documents.

Developing a tool to employ the the cross lingual similarity may need improvement in retrieving data from Cassandra, which includes schema re-structuring to

support storing category posting at parsing time, instead of the complicated joins Spark does. Such tool may need updating the algorithm that gets new data from Wikipedia and update the data stored in the database instead of completely parsing it, in order to keep our database synched with Wikipedia.

For YAGO fact translation, we need to improve object translation, maybe by adding new entries to the property and object maps. We need also to support more YAGO relations. Then as a next step, we need to run the translation against the whole YAGO data to have a complete Arabic knowledge base. Creating unstructured data from that knowledge comes next, which is already covered in the proposed approach but hasn't been implemented yet.

We still think that integrating other knowledge bases, DBpedia for example, with Wikidata and YAGO will have a good impact on the results. So this will be another aspect to implement. We may also make use of existing extraction algorithms to add to these knowledge bases even when comparable data is not available in English, or design new ones for Arabic, to enrich the Arabic structured data in the web.

# Bibliography

- [1] M. L. Littman, S. T. Dumais, and T. K. Landauer, “Automatic cross-language information retrieval using latent semantic indexing,” in *Cross-Language Information Retrieval*. Springer, 1998, pp. 51–62.
- [2] F. Ture, J. J. Lin, and D. W. Oard, “Combining statistical translation techniques for cross-language information retrieval,” in *COLING*, 2012, pp. 2685–2702.
- [3] E. Gabrilovich and S. Markovitch, “Computing semantic relatedness using wikipedia-based explicit semantic analysis,” in *IJCAI*, vol. 7, 2007, pp. 1606–1611.
- [4] P. Sorg and P. Cimiano, “Cross-lingual information retrieval with explicit semantic analysis,” in *Working Notes for the CLEF 2008 Workshop*, 2008.
- [5] L. Abouenour, K. Bouzoubaa, and P. Rosso, “Using the yago ontology as a resource for the enrichment of named entities in arabic wordnet,” in *Workshop on Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages Status, Updates, and Prospects. LREC10 Conference*, 2010, p. 27.
- [6] F. M. Suchanek, G. Kasneci, and G. Weikum, “Yago: a core of semantic knowledge,” in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 697–706.
- [7] M. Yahya, S. E. Whang, R. Gupta, and A. Halevy, “Renoun: Fact extraction for nominal attributes,” in *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014.
- [8] A. T. Freeman, S. L. Condon, and C. M. Ackerman, “Cross linguistic name matching in english and arabic: a one to many mapping extension of the levenshtein edit distance algorithm,” in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of*

*the Association of Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 471–478.

- [9] E. Gabrilovich and S. Markovitch, “Computing semantic relatedness using wikipedia-based explicit semantic analysis,” in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 1606–1611.
- [10] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer *et al.*, “Dbpedia- a large-scale, multilingual knowledge base extracted from wikipedia,” *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2012.
- [11] S. Fox and L. Rainie, “The web at 25 in the us,” *Pew Research Centers Internet & American Life Project*, vol. 27, 2014.
- [12] (2011) Arabic online content: Web metric study. [Online]. Available: [http://www.veecos.net/portal/index.php?option=com\\_content&view=article&id=5997:2011-04-16-08-53-48&catid=42:doctorah&Itemid=180](http://www.veecos.net/portal/index.php?option=com_content&view=article&id=5997:2011-04-16-08-53-48&catid=42:doctorah&Itemid=180)
- [13] Supreme council of information and communication technology, arab digital content, ictqatar website. [Online]. Available: <http://www.ictqatar.qa/output/Page2039.asp/>
- [14] Mohammed bin rashid al maktoum foundation, sawaed programme, mohammed bin rashid al maktoum foundation website. [Online]. Available: <http://www.mbrfoundation.ae/English/Entrepreneurship/Pages/Sawaed.aspx>
- [15] (2010) Escwa. report of the final meeting of the project on promotion of the digital arabic content industry through incubation. [Online]. Available: <http://www.escwa.un.org>
- [16] U. N. Economic and S. C. for Western Asia. Saudi Arabia, “Status of the digital arabic content industry in the arab region,” November 2012.
- [17] K. Darwish and W. Magdy, *Arabic information retrieval*. Now Publishers, 2014.
- [18] L. S. Larkey and M. E. Connell, “Arabic information retrieval at umass in trec-10,” DTIC Document, Tech. Rep., 2006.
- [19] D. W. Oard and A. R. Diekema, “Cross-language information retrieval,” *Annual review of information science and technology*, vol. 33, pp. 223–256, 1998.

- [20] A. Yahya and A. Salhi, “Quality assessment of arabic web content: The case of the arabic wikipedia,” in *10th International Conference on Innovations in Information Technology (INNOVATIONS)*, 2014. IEEE, 2014, pp. 36–41.
- [21] —, “Enhancement tools for arabic web search: A statistical approach,” in *7th International Conference on Innovations in Information Technology*, 2011.
- [22] P. Sorg and P. Cimiano, “An experimental comparison of explicit semantic analysis implementations for cross-language retrieval,” in *Natural Language Processing and Information Systems*. Springer, 2010, pp. 36–48.
- [23] A. Gupta, A. Kumar, J. Gautam, A. Gupta, M. A. Kumar, and J. Gautam, “A survey on semantic similarity measures,” *IJIRST-International Journal for Innovative Research in Science & Technology*, vol. 3, p. 12, 2017.
- [24] G. Majumder, P. Pakray, A. Gelbukh, and D. Pinto, “Semantic textual similarity methods, tools, and applications: A survey,” *Computación y Sistemas*, vol. 20, no. 4, pp. 647–665, 2016.
- [25] W. H. Gomaa and A. A. Fahmy, “A survey of text similarity approaches,” *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13–18, 2013.
- [26] R. Mihalcea, C. Corley, and C. Strapparava, “Corpus-based and knowledge-based measures of text semantic similarity,” in *Proceedings of The 21st National Conference on Artificial Intelligence - Volume 1*, 2006, pp. 775–780.
- [27] P. D. Turney, “Mining the web for synonyms: Pmi-ir versus lsa on toefl,” in *European conference on machine learning*. Springer, 2001, pp. 491–502.
- [28] T. K. Landauer, P. W. Foltz, and D. Laham, “An introduction to latent semantic analysis,” *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.
- [29] R. L. Cilibrasi and P. M. Vitanyi, “The google similarity distance,” in *IEEE Transactions on Knowledge and Data Engineering*, 2007, pp. 370–383.
- [30] T. Chklovski and P. Pantel, “Verbocean: Mining the web for fine-grained semantic verb relations,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.
- [31] A. Budanitsky and G. Hirst, “Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures,” in *In Workshop on Wordnet and other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics*, 2001.

- [32] G. Hirst, D. St-Onge *et al.*, “Lexical chains as representations of context for the detection and correction of malapropisms,” *WordNet: An electronic lexical database*, vol. 305, pp. 305–332, 1998.
- [33] C. Leacock and M. Chodorow, *Combining local context and WordNet sense similarity for word sense identification*. Massachusetts Ave, Cambridge, United States: Massachusetts Institute of Technology, 1998, pp. 265–283.
- [34] Z. Wu and M. Palmer, “Verb semantics and lexical selection,” in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, 1994, pp. 133–138.
- [35] P. Resnik, “Using information content to evaluate semantic similarity in a taxonomy,” in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*. Morgan Kaufmann, 1995, pp. 448–453.
- [36] D. Lin, “An information-theoretic definition of similarity,” in *Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, pp. 296–304.
- [37] A. Panchenko, O. Morozova, and H. Naets, “A semantic similarity measure based on lexico-syntactic patterns,” in *Proceedings of KONVENS*, 2012.
- [38] T. Gottron, M. Anderka, and B. Stein, “Insights into explicit semantic analysis,” in *Proceedings of the 20th ACM conference on information and knowledge management (CIKM11)*. ACM, 2011.
- [39] P. Sorg and P. Cimiano, “Cross-lingual information retrieval with explicit semantic analysis,” in *Working Notes for the CLEF 2008 Workshop*, 2008.
- [40] F. Mahdisoltani, J. Biega, and F. Suchanek, “Yago3: A knowledge base from multilingual wikipedias,” in *7th Biennial Conference on Innovative Data Systems Research*. CIDR Conference, 2015.
- [41] (2019) Category:disambiguation pages. [Accessed 28-March-2019]. [Online]. Available: [https://en.wikipedia.org/wiki/Category:Disambiguation\\_pages](https://en.wikipedia.org/wiki/Category:Disambiguation_pages)
- [42] (2019) Wikipedia:redirect. [Accessed 28-March-2019]. [Online]. Available: <https://en.wikipedia.org/wiki/Wikipedia:Redirect>
- [43] (2019) Wikimedia downloads. [Accessed 27-March-2019]. [Online]. Available: <https://dumps.wikimedia.org/backup-index.html>
- [44] (2019) List of nosql databases. [Accessed 28-March-2019]. [Online]. Available: <http://nosql-database.org/>



- [45] (2019) Apache spark. [Accessed 28-March-2019]. [Online]. Available: <https://spark.apache.org/>
- [46] Wikipedia contributors. (2019) Apache spark — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Apache\\_Spark&oldid=887875725](https://en.wikipedia.org/w/index.php?title=Apache_Spark&oldid=887875725). [Accessed 28-March-2019].
- [47] . (2018) Cluster mode overview. [Accessed 28-March-2019]. [Online]. Available: <https://spark.apache.org/docs/latest/cluster-overview.html>
- [48] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2009, vol. 39.
- [49] (2018) Stanford log-linear part-of-speech tagger. [Date last accessed 20-July-2017]. [Online]. Available: <https://nlp.stanford.edu/software/tagger.shtml>
- [50] B. Santorini, “Part-of-speech tagging guidelines for the penn treebank project (3rd revision),” *Technical Reports (CIS)*, p. 570, 1990.
- [51] (2016) Yago: A high-quality knowledge base. [Date last accessed 20-May-2016]. [Online]. Available: <http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/>
- [52] J. Biega, E. Kuzey, and F. M. Suchanek, “Inside yago2s: A transparent information extraction architecture,” in *Proceedings of the 22nd international conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee, 2013, pp. 325–328.
- [53] (2016) Wikidata. [Date last accessed 20-May-2016]. [Online]. Available: <http://www.wikidata.org/>
- [54] D. Vrandečić and M. Krötzsch, “Wikidata: a free collaborative knowledge base,” *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.
- [55] (2016) Wikidata. [Date last accessed 27-May-2016]. [Online]. Available: <http://wiki.dbpedia.org/>
- [56] (2016) Wikipedia. [Date last accessed 30-May-2016]. [Online]. Available: <http://www.wikipedia.org/>
- [57] Wikipedia Contributors, “Featured articles,” 2018, [Accessed 11-July-2018]. [Online]. Available: [https://en.wikipedia.org/wiki/Wikipedia:Featured\\_articles](https://en.wikipedia.org/wiki/Wikipedia:Featured_articles)

- [58] Wikipedia contributors. (2018) ويكيبيديا:مقالات مختارة. [Accessed 11-July-2018]. [Online]. Available: [https://ar.wikipedia.org/wiki/:\\_](https://ar.wikipedia.org/wiki/:_)
- [59] OPUS: The Open Parallel Corpus. (2018) Global voices parallel corpus. [Accessed 13-July-2018]. [Online]. Available: <http://opus.nlpl.eu/GlobalVoices.php>
- [60] P. Adolphs, M. Theobald, U. Schafer, H. Uszkoreit, and G. Weikum, “Yagoqa: Answering questions by structured knowledge queries,” in *2011 IEEE Fifth International Conference on Semantic Computing*. IEEE, 2011, pp. 158–161.
- [61] E. Kuzey and G. Weikum, “Evin: Building a knowledge base of events,” in *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014, pp. 103–106.

# Appendices

# Appendix A

## Cassandra Create Script

```
drop
KEYSPACE if exists wiki_en;
create
KEYSPACE wiki_en with REPLICATION = { 'class':'SimpleStrategy',
'replication_factor':1 };
USE Wiki_en;
-- article Table --
create
table
article(
wikibase_item varchar,
title varchar,
user varchar,
content TEXT,
size int,
normalized_content_length int,
category_count int,
primary key(
wikibase_item,
title
)
);
-- category Table --
create
table
category(
wikibase_item varchar,
```

```

title varchar,
article_count COUNTER,
primary key(
wikibase_item,
title
)
);
-- info_box Table --
create
table
info_box(
article_wikibase_item varchar,
number int,
content TEXT,
primary key(
article_wikibase_item,
number
)
);
-- article_category Table --
create
table
article_category(
article_wikibase_item varchar,
category_wikibase_item varchar,
primary key(
article_wikibase_item,
category_wikibase_item
)
);
-- category_link Table --
create
table
category_link(
category_wikibase_item varchar,
parent_category_wikibase_item varchar,
primary key(
category_wikibase_item,
parent_category_wikibase_item
)
);

```

```

);
-- dictionary Table --
create
table
dictionary(
word varchar,
frequency counter,
article_count counter,
primary key(word)
);
-- posting Table --
create
table
posting(
word varchar,
article_wikibase_item varchar,
position TEXT,
frequency int,
primary key(
word,
article_wikibase_item
)
);
-- representative_category Table --
create
table
representative_category(
wikibase_item varchar,
primary key(wikibase_item)
);
-- representative_article Table --
create
table
representative_article(
wikibase_item varchar,
primary key(wikibase_item)
);
-- representative_category_link Table --
create
table

```

```
representative_category_link(  
  category_wikibase_item varchar,  
  parent_category_wikibase_item varchar,  
  primary key(  
    category_wikibase_item,  
    parent_category_wikibase_item  
  )  
);
```

# Appendix B

## Wikipedia Featured Articles

- وي, Wii
- ليزا ديل جوكونديو, Lisa del Giocondo
- مايا أنجيلو, Maya Angelou
- هاري ترومان, Harry S. Truman
- بوبي روبسون, Bobby Robson
- ستيف بروس, Steve Bruce
- تاريخ آلات قياس الوقت, History of timekeeping devices
- محمد علي جناح, Muhammad Ali Jinnah
- مرصد هابل الفضائي, Hubble Space Telescope
- نادي برشلونة, FC Barcelona
- كوارك, Quark
- أعرف لماذا يغرد الطائر الحبيس, I Know Why the Caged Bird Sings
- الجبهة الغربية (الحرب العالمية الأولى), Western Front (World War I)



- ماموث صوفي, Woolly mammoth
- إسطنبول, Istanbul
- حمى الضنك, Dengue fever
- نيلسون مانديلا, Nelson Mandela
- استراتيجية سنغافورة, Singapore strategy
- نظرية التعلق, Attachment theory
- ملعب أنفيلد, Anfield
- فلور, Fluorine
- جورج الثالث ملك المملكة المتحدة, George III of the United Kingdom
- إيما واتسون, Emma Watson
- "ماريا أو مظالم النساء (رواية)", Maria: or, The Wrongs of Woman
- حركة تحديد النسل في الولايات المتحدة, Birth control movement in the United States
- دينوسوكس, Deinosuchus
- عبور الزهرة, Transit of Venus
- والت ديزني, Walt Disney
- هيدروجين, Hydrogen
- تصلب متعدد, Multiple sclerosis
- أصل الأنواع, On the Origin of Species
- حوسبة متوازية, Parallel computing

- المجموعة الشمسية, Solar System
- الانفجار العظيم, Big Bang
- أسطورة إيزيس وأوزوريس, Osiris myth
- ختان الإناث, Female genital mutilation
- تيتانيا (قمر), Titania (moon)
- انحياز تأكيدى, Confirmation bias
- لاىكا, Laika
- تريسيراتوبس, Triceratops
- مقتل محمد الدرة, Muhammad al-Durrah incident
- بئر العزلة (رواية), The Well of Loneliness
- ملعب أولد ترافورد, Old Trafford
- متلازمة الكبدية الكلوية, Hepatorenal syndrome
- زحل, Saturn
- أوليفيا مانينغ, Olivia Manning
- جورج الثاني ملك بريطانيا العظمى, George II of Great Britain
- فرسان الهيكل, Knights Templar
- بوبروبيون, Bupropion
- فصام, Schizophrenia
- نادى أرسنال, Arsenal F.C.
- توقيت صيفى, Daylight saving time

# Appendix C

## Translated News Articles

- طاقة نظيفة للمستقبل, Clean Ennergy for Tomorrow
- فرنسا: مجلس الدولة الفرنسي يعتبر حظر النقاب مخالفاً للدستور الفرنسي, France: French State Council considers the ban on the Niqab as unconstitutional
- سعر النفط لأعلى مستواياته منذ عام ونصف, Crude oil price hits 18-month high despite strong dolla
- خطوة من مشرعي بلجيكا نحو حظر النقاب في الأماكن العامة, Belgium lawmakers take step towards outlawing niqab in public space
- باكستان: الأمم المتحدة تتهم السلطات بالتقاعس عن حماية بوتو والتحقيق في مقتلها, Pakistan: UN accuses authorities of failing to protect Benazir Bhutto and to investigate her death
- حماس تطلب من وقف إطلاق الصواريخ على إسرائيل و مشعل يؤكد اتخاذ كافة الإجراءات لمنع ذلك, Hamas looks to stop rocket attacks on Israel as Moscow patience runs out
- صربيا تقرر قرارا تاريخيا يدين مجزرة صربنتشا سنة ١٩٩٥, Serbia passes landmark resolution condemning 1995 Srebrenica massacre
- مقال عن اهم المعالم السياحية في مصر مترجم باللغة الإنجليزية, Article about Tourism Sites in Egypt
- أسامة بن لادن في الفايس بوك, Osama bin Laden on Facebook
- سلسلة غارات من الطيران الإسرائيلي على قطاع غزة, Israeli air force launches series of raids on Gaza Stri

- البحرين تضع مسودة قانون يجرم السحر والشعوذة , Bahrain drafts a new law criminalizing sorcery and witchcraft
- أساتذة سعوديون يشتكون من حرمانهم من الرواتب والعقود , Saudi professors still waiting for salaries and contracts
- الانتخابات العراقية: الصدر يستفتي لاختيار رئيس الوزراء والحكيم يرجح كفة علاوي , Iraqi elections: Al-Hakim advocates Allawi while al-Sadr holds referendum to elect prime minister
- انشقاق عالم نووي إيراني ليصبح عميلاً في الاستخبارات الأمريكية , Iranian nuclear scientist defects to the US as CIA agent
- جنبلاط يمدح الرئيس السوري في زيارة له , Druze MP Walid Jumblatt praises Bashar al-Assad's Lebanon stance
- أبوظبي تسعى لمنح بعد جديد للسياحة البيئية في المنطقة , Abu Dhabi looks to add a new dimension to eco-tourism in the region
- السودان: المعارضة تجتمع اليوم لحسم موقفها من الانتخابات , Sudan: Opposition meeting today to decide about its position from the elections
- المغرب تحمي البيئة بمشروع جديد لاستغلال الطاقة الشمسية , Morocco protects the environment with new solar energy project
- السناتور كيري : دور سورية هام جدا لحياء جهود السلام , Senator Kerry describes Syria's role as pivotal to revive peace
- أمغامرات سلوى، سلسلة رسومات هزلية لمكافحة التحرش الجنسي , Lebanese campaign group IndyACT launches comic series to fight sexual harassment
- بدء تصوير مسلسل أذاكرة الجسد، أحد أشهر الروايات العربية , Arab world's 'most celebrated novel' to be turned into 30-part TV series
- إسرائيل تتهم سوريا بتسليح حزب الله بالصواريخ , Israel accuses Syria over Hezbollah missiles
- الجزائر تعتقل عميلاً للموساد لاثامه بدخول البلاد بجواز سفر أسباني مزور , Algeria detains alleged Mossad agent with fake Spanish passport

- ٢٠٠, US\$200 million to Improve Education for Jordanian and Syrian Refugee Students
- ميركل في زيارة حساسة لتركيا و أردوغان يستبق الزيارة برفض العقوبات على ايران, Merkel's visit to Turkey preceded by Erdogan's rejection of Iran sanctions
- رافات وحقائق مرض ميزوثيليوما الرئوي, The Myths and Facts of Mesothelioma
- ٧, 7 Great Steps to Have the Perfect Makeup
- البنك العربي, Arab Bank
- هاكوب يكلف منذ ٤٠ عاماً للحفاظ على الخزف الأرميني في القدس, Armenian Ceramics Artist Keeps Ancient Craft Alive in Jerusalem
- الفن هو صورة صحية للتعبير, Art for Your Health
- هجمات الحادي عشر من ايلول, 9/11 Attacks
- هل من الممكن أن يكون هذا نهاية السبام؟, Could this be the end of Spam?
- تدابير خارجة عن القانون تحت الضوء في فلسطين, Extrajudicial measures under the spotlight in Palestine
- بعد تسلّم السلطة الفلسطينية معابر قطاع غزة هل سيتوقف العمل بنظام جفي؟, Israel to lift tight controls over Gaza reconstruction materials
- إم بي ثري.. جنون عالمي, MP3 - a worldwide mania
- فلسطين ستفتتح بنكها المركزي قبل نهاية عام ٢٠١٧, Palestine to Get its Own Central Bank
- مدى نجاح المصالحة بتوحيد القوانين بين غزة والضفة, Parallel legal systems pose threat to Palestinian unity
- قطر للغاز, Qatar Petroleum
- سوريا: سنة من الاحتجاجات والتمرد, Syria's year of protest and insurrection
- الحرب الباردة, The Cold War

- مكتبة الكونجرس, The Library of Congress
- الأمم المتحدة, The United Nations
- واشنطن تقترح مفاوضات مباشرة على إسرائيل مقابل تجريد الاستيطان ٤ أشهر, Washington calls for 4-month Israeli settlement freeze to kick start direct negotiations
- ما هو التأمين الصحي؟, What is Health Insurance?
- وزارة الحكم المحلي توقف العمل بقانون توزيع الأراضي الحكومية وحساس ترفض, Will end of land-for-pay program kill Palestinian reconciliation?
- الحرب العالمية الثانية, World War II

## Appendix D

### Bilingual Tweets

- جلالة الملك عبدالله الثاني يتلقى اتصالا من جلالة الملك حمد بن عيسى آل خليفة، ملك مملكة البحرين، تناول العلاقات الأخوية الراسخة التي تجمع البلدين والشعبين الشقيقين، والحرص على توطيدها في المجالات كافة، ومواصلة التشاور والتنسيق إزاء مختلف الشئون المشتركة بين البلدين والشعبين الشقيقين.  
His Majesty King Abdullah II receives a phone call from King Hamad bin Isa Al Khalifa of Bahrain, and discusses the deep-rooted, brotherly ties between Jordan and Bahrain, and keenness to strengthen them
- جلالة الملك عبدالله الثاني يتلقى اتصالا من الرئيس الفلسطيني محمود عباس، جرى خلاله تناول آخر التطورات على الساحة الفلسطينية، وذلك في إطار التنسيق والتشاور المستمرين بين الجانبين الأردني والفلسطيني.  
His Majesty King Abdullah II receives a phone call from Palestinian President Mahmoud Abbas, as part of ongoing Jordanian-Palestinian coordination, and discusses the latest developments in the Palestinian Territories Jordan
- جلالة الملك عبدالله الثاني يكلف الدكتور عمر الرزاز بتشكيل حكومة جديدة الأردن  
His Majesty King Abdullah II tasks Omar Razzaz with forming a new government
- الذكاء الاصطناعي لن يتطابق أبدا مع عيش التجربة. لا يوجد قوة خوارزمية ستساوي في يوم قلب الإنسان  
Artificial intelligence is no match for lived experience. No algorithms power will ever equal a human heart

- عندما نتعامل مع الرموز الرقمية التي تعيد تعريف حياتنا في الواقع، هنالك رموز أقدم ما زالت تتطلب اهتمامنا. رموز انسانية مثل التعاطف والرحمة والتفاهم، والتي يمكن  
As we grapple with the digital codes that are redefining our lives in real time, there are older codes that still require our attention. Human codes like empathy compassion understanding
- مقتطفات من كلمة جلالة الملك عبدالله الثاني أمام طلبة الجامعات المشاركين في  
Excerpts from His Majesty King Abdullah II's speech to the students of the World Class The Hague programme in the Netherlands
- مع أهل الكرم والضيافة في قرية الرميمين في البلقاء الذين استقبلوا المشاركين في  
With the generous and hospitable people of Rmeimeen Village in Al Balqa, who warmly welcomed Jordan Trail hikers today
- ونحن نستقبل عيد الميلاد المجيد، أهنيئ إخواننا المسيحيين وأتمنى لهم الخير والبركة، وأتمنى لوطننا في العام الجديد مزيداً من الازدهار والتقدم بسواعد أبنائه وبناته وإرادتهم الصلبة، التي تثبت دائماً أنها أقوى من كل التحديات. عيداً مجيداً وعاماً  
Sending wishes of blessings and joy to our Christian brothers and sisters on Christmas. May 2018 bring further prosperity and progress to all Jordanians, who continue to conquer challenges with their firm resolve. A blessed Christmas and a happy New Year to Jordan
- قالن: الهدف من ذلك (اغلاق ٧ مساجد في النمسا) تحقيق مكسب سياسي من خلال اقضاء  
Turkish presidential aide says Austrias aim is to obtain political gain by marginalizing Muslim communities
- #Egypts Sisi , #مصر.. السيسي يكلف وزير الإسكان بتشكيل حكومة جديدة  
orders housing minister to form new govt
- #أمريكا تقدم ٥.٣ \$ مليار سنوياً لدعم الفصل العنصري في #إسرائيل.. مبلغ يكفي



The United States gives \$3.5 billion to Apartheid Israel every year, enough to feed half the worlds hungry children.

- أكدت دراسات بعض الباحثين أن الشخص في شمال #أمريكا يتعرض لـ ٣٣٣ إعلان يومياً أي

Researchers have shown that every day that average person in North America is exposed to 330 ads, thats one ad every 3 minutes.

- برغم أن ٩٩ % من الذهب موجود في أراضي #أفريقيا، إلا أن ٤ أفارقة فقط (بينهم

There are only 4 African billionaires in the world, Oprah included, even though resource rich Africa has 90% of the worlds gold

- أظهر إستطلاع رأي أُقيم على متابعي قناة فوكس نيوز أن ٢٨ % منهم يعتقدون أن A survey of Fox News viewers has found that 82% of them think that Palestinians spend more on weapons than Israel.

- أول تغريدة من الفضاء: من المدار: كان إنطلاقاً رائعاًنا بصحة ممتازة وأعمل بكد  
From orbit: Launch was awesome!!  
I am feeling great, working hard, & enjoying the magnificent views, the adventure of a lifetime has begun!

- قناة #الجزيرة في #قطر غطت في نشراتها احتجاجات وول ستريت في #أمريكا أكثر من

Aljazeera has done more stories on the Wall Street protests than CNN, Fox News and MSNBC combined.

- يهـار: آباء يتسلقون جدران المدرسة للوصول إلى نوافذ قاعات الإمتحانات لرمي أوراق

Bihar: Parents climb walls to reach the window of exam halls to throw answer chits to their children #WallOfShame

- إن جزءاً من الإجابة يكمن في إعادة تصوّر التعليم في إعداد الأجيال الصاعدة  
للازدهار في هذه المرحلة الجديدة... ويعني ذلك تسليحهم للتميز بكل الطرق التي لا يمكن

The answer partly lies in reimagining #education in preparing rising generations to thrive in this new time... it means equipping them to excel in all the ways machines cannot #EBRDam , للآلات التفوق فيها

- اليوم نتذكر بألم كل طفل راح ضحية الحرب في سوريا، والعائلات التي تبحث عن ملجأ، والمنازل والأحياء التي كانت يوماً تعني الكثير لأهلها. ومع ذلك ما يجب أن يفجعنا بعد Today we mourn , سبع سنوات، هو فشل المحاولات الدولية لإنهاء النزاع في سوريا. the children who lost their lives in the Syrian war, the families seeking refuge, and the homes and streets that once meant so much to Syrians. Yet what we must mourn more ardently after seven years, is the futility of the worlds attempts to end the conflict.
- قبل ٣ سنوات أطلقنا منصة إدراك - للتعليم الإلكتروني ونجحنا بالوصول لأكثر من مليون ونصف متعلم. ونحن على وشك إدراك حلم جديد بإطلاق منصة إلكترونية تعليمية باللغة العربية مخصصة لطلبة المدارس والمعلمين بالشراكة مع جوجل دوت أورغ. فخورة بهذه الخطوة I launched the Edraak online learning platform , التي ستثري التعليم المدرسي 3 years ago, which now has over 1.5 million adult learners. In partnership with @Googleorg, Edraak is about to take an incredible step forward with the launch of an online learning platform for Arab school children & their teachers
- رئيس الوزراء الفلسطيني د. رامي الحمد الله يشارك في الإفطار الخيري لأسر وذوي الشهداء والأسرى، ويؤكد على أن حقوق أهالي وعائلات الشهداء مصانة رغم كافة التحديات، ويطالب بأوسع اصطفاة دولي حول دعوة الرئيس محمود عباس لعقد مؤتمر دولي يقر آلية دولية Prime Minister Dr. Rami Hamdallah , متعددة الأطراف لرعاية عملية السلام takes part in a Ramadan Iftar dinner in honour of the families of martyrs and prisoners. The Prime Minister calls for an international support for the peace initiative of H.E. President Mahmoud Abbas.
- رئيس الوزراء د. رامي الحمد الله يستقبل وفدا من رجال الدين المسيحيين برئاسة راعي كنيسة الروم الارثوذكس في رام الله الارشمندريت الياس عواد. ويؤكد اعتنازه بوحدة

PM Dr. Hamdallah received a delegation of Christian clergy chaired by the Greek Orthodox Church patron in Ramallah, Archimandrite Elias Awwad. H.E Al Hamdallah affirmed his pride in the unity of the communal fabric among all religious communities in Palestine

- ما هو إحساسك عندما يحصل هذا الامر في بلدك..هذه القصص حقيقية وليست  
If This Happened in Your Country, What Would You Feel , خيالية.. #فلسطين  
? These stories are not imaginary..#palestine
- تتوجه بالشكر للأرجنتين وليونيل ميسي على إلغاء المباراة الودية مع إسرائيل،  
وتتمنى للأرجنتين كل التوفيق في كأس العالم، وهذه رسالة بليغة للفيفا لطرده إسرائيل حتى  
Palestinians thank @Argentina & Capt. Messi for canceling their World Cup warmup match w Israel. All the best 4 the #WorldCup! @FIFAcorn must take note. Its time 2 #RedCardIsrael 4 its violations against Palestinian football & disregard 4 FIFA statutes
- جلالة الملك عبدالله الثاني يتلقى اتصالاً من خادم الحرمين الشريفين الملك سلمان بن عبدالعزيز آل سعود جرى خلاله استعراض العلاقات التاريخية الراضة بين البلدين الشقيقين، ومستجدات الأوضاع في المنطقة، والتأكيد على مواصلة التنسيق والتشاور إزاء  
His Majesty King Abdullah II receives a phone call from Custodian of the Two Holy Mosques King Salman bin Abdulaziz of #Saudi Arabia and discusses historical ties, regional developments, and importance of maintaining coordination on various issues #Jordan
- من زيارة سمو الأمير الحسين بن عبدالله الثاني، ولي العهد، اليوم للاطمئنان على  
From HRH , صحة عدد من مصابي الأمن العام وقوات الدرك في مدينة الحسين الطبية  
Crown Prince Al Hussein bin Abdullah IIs visit to check on injured public security and gendarmerie personnel receiving treatment
- نتطلع إلى استضافة هشام القرق في #سلسلة الرواد غداً من الساعة ١٠:٠٠ مساءً -  
٣٠:١١ مساءً، حيث سيشاركنا خبراته القيمة في مجال ريادة الأعمال تحت عنوان هل شركتك

.جاهزة للاستثمار؟, We are looking forward to having Hisham Al Gurg share his valued experience on entrepreneurship and getting your company ready for investment. Join us in this #Pioneer\_series session tomorrow from 10:00 p.m. 11:30 p.m.

- أثناء استضافتنا لمعالي مريم المهيري، وزيرة دولة للأمن الغذائي المستقبلي، في . During H.E. Mariam Almuhairi, Minister of state for Future food security, fruitful session on Future shaping the national food security landscapes
- يعد صوت المدفع من أهم العادات الرمضانية باعتباره الوسيلة الوحيدة للإعلان عن موعد الإفطار لحظة مغيب الشمس قبل ظهور المكبرات الصوتية اليوم، أصبح إطلاق المدفع . Nothing signifies the holy month of Ramadan quite like the sound of cannon fire at sunset! Today, the cannon is followed as one of many traditions of Ramadan. #VisitQatar #Ramadan-InQatar
- على بعد عشرين دقيقة من قلب الدوحة؛ يكمن الملاذ المناسب والخلاب! مع كل المرافق والخدمات الراقية والأنشطة الشاطئية الممتعة التي ستبهركم سواء كانت هذه زيارتكم الأولى Just 20-minutes away from downtown #Doha lies a lush oceanfront oasis! Ideal this time of year, with its wide range of amenities, services & water sport activities, whether its your first time to #VisitQatar or youre planning a staycation this weekend
- كشف الموسم التاسع من برنامج نجوم العلوم أن بناء علم أفضل من شأنه أن يكسر الحواجز السياسية. يجمع برنامج نجوم العلوم ما بين رواد أعمال شباب وطموحين من جميع أنحاء المنطقة، لطرح حلول ملموسة لبعض من أكثر المشكلات إلحاحًا في العالم . The ninth season of Stars of Science revealed that building better science can break down political barriers. Stars of Science brings together young entrepreneurs from around the region to create innovative solutions for various challenges. #QatarFoundation #QatarStronger #QatarMovingForward

- جامع الدولة ، أو المعروف بإسم جامع الإمام محمد بن عبد الوهاب، هو أحد الأماكن التي يجب زيارتها! ننصحكم بالتخطيط لزيارته مساءً لتجربة لا تنسى وكي تستمتعوا بتصميمه الخلاب الذي يضفي الكثير من الهدوء والسكينة الى نفوس زائريه، الامر الذي سيتمنحكم
- The Grand Mosque, or otherwise known as Imam Muhammad ibn Abd al-Wahhab Mosque, is one of the must-sees when you #VisitQatar! Ideally plan a visit at night to be swept away by the space, lights, design and tranquility that will bestow you immediately upon stepping in!
- #UAE Releases دولة الامارات تطلق من دافوس تقرير استشراف المستقبل العالمي Global State of the Future Report at the World Economic Forum Davos
- لا يمكنك زيارة العين من دون تجربة تسلق جبل حفيت الذي يعد أعلى قمة #فيأبوظبي You cant visit Al Ain and not explore its rocky mountain, Jebel Hafeet. Ready to look beyond the horizon and up to the highest peak #InAbuDhabi?
- نشر كل من تقدّم بالمشاركة في مسابقة أبوظبي من خلال عيونكم ٢٠١٨. سيتم الإعلان عن قائمة المرشحين في شهر يوليو. سنوافيكم قريباً بمزيد من المعلومات عن We would like to thank everyone who entered Abu Dhabi Through Your Eyes 2018. The shortlist will be announced in July. Watch this space for more information on the new Peoples Choice Award!
- وزير الدفاع التركي يقول إن إيران عرضت مساعدة بلاده في العمليات العسكرية التي Turkish Defense minister said Iran has offered to help his country in its military operations in northern Iraq , تقوم بها في شمال العراق
- هل يمكن أن نصيب أنفسنا بأذى أثناء قذف الكرة أو القفز بالحبل أو اللعب بكرة الحبل أو عند الدخول إلى المساحات الضيقة أو المشاركة في سباقات الرالي أو ركوب Can we hurt الأرجوحة الشبكية؟ سنعرف ذلك في حلقتين متتاليتين من علم البسطاء ourselves while throwing a ball, jumping rope, playing rope reel or getting

into cramped spaces or participating in rally races or riding a hammock? Well figure it out in two successive episodes of simple #علم البسطاء #تجارب #علوم Science.

- ٢٢ ألف بولندي قتلوا خلال الحرب العالمية الثانية في جريمة إنسانية كبرى، من 22, 000 Poles killed during the Second World War in a major humanitarian crime, who killed them and who covered up their crime? More to the last witness: #الحزيرة الوثائقية
- قد لا تأخذ هذا الفيلم على محمل الجد، ولكنه سيجعلك تدرك أنك لم تذوق طعم الطماطم من قبل، إذ سيأخذك في رحلة داخل الطماطم حرفياً، لتعلم لماذا يتصرف الناس بحنون في You may not take this film seriously, but it will make you realize that youve never tasted tomato before, taking you on a trip inside the tomato literally, to learn why people behave madly in tomato festivals in Italy and Spain.
- يعد المسلمون فيها الغالبية حيث يمثلون نحو ٤٠ ٪ من تعدادها السكاني، انطلقت فيها رصاصة الحرب العالمية الأولى، وتعرضت للحصار من الجيش اليوغسلافي وقتل أثناء الحصار ١١ ألفاً، وقد تميزت بتاريخها الغني بالتنوع الديني لذلك عرفت باسم قدس أوروبا فهل عرفتوها Muslims are the majority, accounting for about 40 percent of the population, where the First World War shot was launched and the siege was imposed by the Army Yugoslavian killed during the siege 11, 000, and was characterized by its rich history of religious diversity so I knew the name of The Holiness of Europe.
- شاهد الفيلم الوثائقي رمضان في الجزائر في مقهى الجزيرة الإعلامي Watch the documentary film Ramadan in Algeria at Al Jazeera Media Caf
- يعد المجتمع السعودي من المجتمعات الشابة؛ إذ بلغ عدد سكان المملكة تحت سن ٢٤ عاماً نحو ٤٧ ٪. يتألف المجتمع السعودي من ثلاث فئات وهي البادية والريف والحاضرة. يمثل البدو ٢١,٧٧ ٪ والريفيون ٢٦,٨٧ ٪ بينما تمثل فئة الحضر نحو ٥١,٣٦ ٪ The Saudi community is a young society; The kingdoms population under

24 years of age is about 47%. The Saudi society consists of three categories, namely the Badia, the countryside and the present. Bedouins represent 21, 77% and the rural population is 26, 87%, while the urban group is approximately 51, 36%

- هل تعلم أن الكونغو قادرة على توفير الغذاء لثلث العالم كونها تمتلك ٨٠ مليون هكتار من الأراضي الخصبة. ولكنها حالياً تعتبر أفقر دولة في العالم نتيجة الحروب الأهلية التي عصفت بالبلاد. هل تعلم أن الكونجو تعتبر ثاني أهم احتياطي عالمي من النحاس الأهلية التي عصفت بالبلاد.
- Did you know that the Congo is able to provide food for one third of the world as it possesses 80 million hectares of fertile land. But it is currently the poorest country in the world as a result of the civil wars that have ravaged the country. Did you know that the Congo is the second most important global reserve of Copper Navigator on Earth, 10%.
- في مثل هذا اليوم عام ١٩٩١، تم حل الاتحاد السوفيتي رسمياً وتم تفكيكه إلى ١٦ دولة عقب إصدار مجلس السوفييت الأعلى للاتحاد السوفيتي إعلان تم فيه الاعتراف باستقلال الجمهوريات السوفيتية السابقة، وإنشاء رابطة الدول المستقلة لتحل محل الاتحاد السوفيتي.
- , On this day in 1991, the Soviet Union was formally dissolved and dismantled to 16 countries following the promulgation by the Soviet Union of the USSR of a declaration recognition of the independence of the former Soviet republics and the establishment of the Commonwealth of Independent States (CIS) to replace the Soviet Union.
- ي ١٤ ديسمبر الماضي توفي الرسام الأمريكي الشهير بوب جيفنز عن عمر يناهز ٩٩ عام، وهو الرسام الذي شارك في تصميم شخصيات كرتونية محببة لدى الكثيرين مثل الأرنب باغز باني
- On December 14, the famous American painter Bob Givens died at the age of 99, a painter who participated in the design of likable cartoon characters I have a lot of people like Bugs Bunny, Tom and Jerry, Sailor Popeye, and Daffy Duck.
- انتحار الجنرال الكرواتي البوسني سلوبودان برالجاك أثناء استماعه لحكم المحكمة الجنائية الدولية ليوغوسلافيا السابقة فيما يخص الحرب ضد السكان البوسنيين

The suicide of Bosnian Croat general Slobodan Praljak during his hearing of the ruling of the International Criminal Tribunal for the former Yugoslavia in relation to the war against the Bosnian Muslim population in some 30 municipalities in Bosnia and Herzegovina.

- غوغل يحتفل بالذكرى الـ ٨٠١ لميلاد أشهر فنانة تركية في مجال النحت على السيراميك  
Google celebrates the 108th anniversary of the birth of the most famous Turkish artist in the field of ceramic
- صرحت المحكمة العليا بأن حكم صهر الملك الإسباني قد صدر يوم الثلاثاء بالسجن لمدة خمس سنوات وعشرة أشهر بتهمة حيازة ملايين اليورو في قضية أخرجت العائلة المالكة.  
, The Supreme Court ruled that the Spanish kings smelting verdict was issued on Tuesday with five years and 10 months in prison for the possession of millions of euros in a case that embarrassed the royal family.
- كل المساعدات العسكرية الأميركية للجيش اللبناني (كما كل المساعدات والمبيعات العسكرية الأميركية لكل الجيوش العربية) تمرّ عبر اللوبي الاسرائيلي وهو يوافق عليها بشرط: (١) عدم الإخلال بتفوق العدو الاسرائيلي النوعي. (٢) ان لا تُطلق رصاصة أميركية.  
All U.S. military assistance to the Lebanese Army (as all U.S. military assistance and sales to all Arab armies) passes through the Israeli lobby and is approved on condition that: (1) Without prejudice to the superiority of the Israeli enemy in qualitative. 2) not to fire one American shot against the enemy.
- اعتادوا أن يشاهدوا طائرات من نوع آخر، واليوم هم يخلقون بطائراتهم التي تعاونوا  
They used to see a different type of planes, but today they fly their own kites and spread the joy over the camp sky!
- لا شيء يضاهي روعة مذاق القهوة الساخنة والساندويش قبل السفر لدى ثمدن ذا د ث  
Nothing quite like a hot coffee and a sandwich at the Camden Food Co before your flight. Located in T3 departures



- Discover all the banking services available at Abu Dhabi Airport , اكتشف كافة الخدمات المصرفية المتوفرة لدى مطار أبوظبي الدولي.
- كجزء من الحملة التوعوية عن أمن المعلومات أطلقت مطارات أبوظبي مبادرة أمن المعلومات للمسافرين والموظفين , As a part of information security awareness Abu Dhabi Airport Launches Information security initiative to its passengers and employees
- ٥٥ % من سكان قطاع غزة يعانون من الاكتئاب، بسبب الحصار الإسرائيلي المفروض على القطاع منذ ١٢ عاما والذي تخلله ثلاثة حروب ارتقى خلالها مئات الشهداء وأصيب الآلاف . 55% of the population of the Gaza Strip suffers from depression, due to the Israeli siege of the strip 12 years ago, which was marked by three wars in which hundreds of martyrs were raised, thousands were injured and infrastructure was heavily destroyed.
- شركة اشعم صيخ أعلنت انها ستوفر في عام ٢٠٢٢ اول رحلة تجارية للفضاء عبارة عن اجازة ل ١٠ ايام في مركبة خاصة متصلة بمحطة الفضاء الدولية وتكلف ٥٥ مليون دولار للشخص , Axiom Space has announced that in 2022 it will provide the first commercial space flight, a 10-day vacation in a private vehicle connected to the ISS and costing \$55 million per person
- انفجر خزان الأكسجين الذي كان يحمله خلال رحلة التسلق.. فكيف تمكن من البقاء على قيد الحياة؟ تعرفوا إلى قصة بين فوغل الذي وصل إلى أعلى نقطة على سطح الأرض , The oxygen tank that he carried during the climbing trip exploded. How could he survive? You know a story between Vogel, who reached the highest point on Earth
- في أواخر ستينيات القرن الماضي كانت بي إم دبليو تطمح لدخول السوق الأمريكية. وقامت بابتكار سيارة السيدان الرياضية ٢٢ ٢ لمواكبة السوق في ذلك الوقت والتي أصبحت الخيار الأمثل لسباق السيارات. لم تتوقف بي إم دبليو عند ذلك، وبدأت الشركة عام ٣٧٩١ In the late 1960s, BMW was aspiring to enter the American market. I invented Mr. 2002 sports car to keep up with the market at the time, which has become the perfect choice

for the car races. The BMW did not stop at that, and in 1973 the company began to produce a turbo-fueled model of car 2002 .

- بدءاً من العام ٢٢ ٥١ وحتى الآن قمنا بتوليد أكثر من ٩.٦ مليار طن من النفايات البلاستيكية، تم إعادة تدوير ٩% منها فقط وتم حرق ٢١% بينما تراكم ٩٧% في مدافن النفايات أو البيئة الطبيعية. Starting in the year 2015 and so far we have generated more than 6.9 billion tons of plastic waste, only 9% of which have been recycled and 12% have been burned while the accumulation of 79% in landfills or natural environment.
- اذا كانت الساعة الآن ٨:٠٠ مساءً في القاهرة من يوم الأربعاء فستكون ١٠:٠٠ صباحاً من يوم الأربعاء في الآسكا الأمريكية ولكنها ستكون ٦:٠٠ صباحاً من يوم الخميس في تشوكوتكا الروسية على الرغم أن المسافة قصيرة جداً بين جزيرة ديوميد الكبرى الروسية If the time now is 8:00 pm in Cairo from Wednesday, it will be 10:00 a.m. Wednesday in Alaska, but it will be 6:00 a.m. from Thursday in Chukotka, Russian although the distance is very short between the Russian Grand Diomed Island and the American Lesser Deomed Island Less than 4 km
- يرجع السبب وراء موت مليون فردٍ (٧.٧ %) في الاتحاد الأوروبي إلى زيادة الوزن. وفي المتوسط، تُخَفِّض السمنة أو البدانة من متوسط العمر المأمول من ستة إلى سبعة أعوام. The reason for the death of 1 million people (7.7 %) in the EU is to increase weight. On average, obesity or obesity from the life expectancy is reduced from about six to seven years.
- في مثل هذا اليوم عام ١٩٩١، تم حل الاتحاد السوفيتي رسمياً وتم تفكيكه إلى ١٦ دولة عقب إصدار مجلس السوفييت الأعلى للاتحاد السوفيتي إعلان تم فيه الاعتراف باستقلال الجمهوريات السوفيتية السابقة، وإنشاء رابطة الدول المستقلة لتحل محل الاتحاد السوفيتي. On this day in 1991, the Soviet Union was formally dissolved and dismantled to 16 countries following the promulgation by the Soviet Union of the USSR of a declaration recognition of the independence of the former

Soviet republics and the establishment of the Commonwealth of Independent States (CIS) to replace the Soviet Union.

- الجمعة السوداء (ملحك زردى) هو اليوم الذي يأتي مباشرة بعد عيد الشكر وعادة يكون في نهاية شهر نوفمبر من كل عام، ويعتبر هذا اليوم بداية موسم شراء الهدايا، في هذا Black Friday , اليوم تقوم أغلب المتاجر في معظم دول العالم بتقديم عروض وخصومات كبيرة Friday is the day that comes immediately after Thanksgiving and usually be at the end of November each year, and today is the beginning of the gift-buying season, on this day most stores in most of the world are making large offers and discounts
- 3 killed and 8 injured by attack on members of the ruling party in Turkey , مقتل ٣ وإصابة ٨ بهجوم على أعضاء في الحزب الحاكم في تركيا
- What is the longest or shortest fasting period you have done? And where were you? , ما هي أطول أو أقصر فترة صيام قمت بها؟ و أين كنت؟
- بمناسبة حلول عيد الفطر المبارك تتقدم أسرة السفارة الاردنية في واشنطن بأحر التهاني والتبريكات سائلين الله عز وجل ان يعيده علينا وعليكم باليمن والبركات , The Embassy of Jordan in Washington DC extend their warmest wishes on the occasion of Eid al-Fitr