Bilingual Summarization of English and Arabic Genetic Diseases Texts

Zainab Almugbel
Lecturer, Computer Science Department, Community College,
Imam Abdulrahman Bin Faisal University,
P. O. Box 1982, Dammam, Saudi Arabia
ORCID: https://orcid.org/ 0000-0003-4570-7088
zhalmugbel@iau,.edu.sa

Abstract

Health Literacy aims at empowering patients to take better decisions about their health. The quality of Health Literacy for patients with genetic diseases can be enriched via facilitating the bilingual retrieval of summaries about the genetic diseases texts from the net. This paper proposes helps translator to achieve this task by utilizing NLP and Recurrent Neural Network (RNN) techniques for two tasks: generating abstractive summarizations and making Arabic-English translation. Both summarization and translation tasks require training sets that can be built from English summaries corpus and Arabic-English parallel corpora. The English summaries corpus is built from Orphadata while the parallel corpora is built from Wiki articles. The corpus is utilized for generating the English summaries from the Wiki articles, and the corpora is utilized for translating these summaries into Arabic. This paper defines the research problem. Then, it investigates a set of objectives to solve the problem. After that, it presents a literature review of the tasks in the objectives. Finally, it discusses the proposed solution for the problem from the following aspects: the required corpora, the system architecture, the RNN memory cell components architectures, the proposed software for the implementation, and the system evaluation.

Keywords: NLP; medical texts; RNN;

1. Introduction

Genetic disease (disorder) is a disease that is caused by a change in a gene; such disease is un-curable, and affects every cell in the body[1]. Many approaches may exist to control the disease's signs, symptoms, and complications, such as dietary changes[2]. For this reason, educating patients about their diseases is crucial. Although the necessity of such task, healthcare professionals may not have time for it [1]. Therefore, employing computer aided approaches are helpful. The net is a valuable corpus that can be utilized to educate patients about their illness [2]. This research focuses on improving the quality of translated Health education for patients with genetic diseases. It proposes combining a corpus approach with Deep Learning techniques to retrieve data that is related to a specific genetic disease. The retrieved data is bilingual, and consists of three main elements: 1. The disease title, 2. a brief summary about the disease that explains the

[1] https://www.yourgenome.org/facts/what-is-a-genetic-disorder
[2] https://ghr.nlm.nih.gov/primer/consult/treatment

severity of a disease and medical terms definitions. 3. a list of symptoms with their definitions. This work attempts to answer the following research question: Can medical related data resources and Deep Learning techniques be utilized to enrich the health literacy with simplified genetic diseases' summaries?
It can be answered via investigating the following set of objectives:

1. build bilingual (Arabic-English) medical parallel corpora. This corpora is useful for the Arabic-English translation task. It is based on Wikipedia articles[3] that are about rare genetic diseases.
2. Build an English summaries corpus that contains high quality data about the diseases from Orphanet[4]. This corpus is utilized to generate English summaries from Wiki articles.
3. Use NLP and Deep Learning techniques to learn how to generate an abstractive summary, where Wikipedia corpus is the input and the Orphdata's summary is the output.
4. Develop an Arabic version of the English summaries corpus (disease title, summary, symptoms) using the medical parallel corpora.
5. Apply semantic similarity methodologies to retrieve bilingual summaries that has similar/related entities to the user's inquiry.
6. Evaluate the system performance on summarization and translation tasks.

These objectives are further explained in the next sections, namely Section 3 and 4. The rest of the research is organized in four sections. The literature review is stated in Section 2; then, the data consideration is discussed in Section 3; finally, the research methodology is proposed in Section 4.

2. Related Work

This section presents the previous research that are related to this research topic. It is categorized into two main subsections: medical texts abstractive summarization, and semantic similarity.

2.1 Abstractive Summarization

Abstractive summarization aims at creating summaries that may not include sentences from the source. Several approaches have been proposed to create this type of summarization. They can be categorized into three types: various approaches, graph-based approaches, and neural network-based approaches.

2.1.1 Various Approaches

This part includes the early attempts for summarizations and some surveys. It presents the related-research which does not depend on a specific technique. The research [3] proposes creating two lists. A list stores key concepts, and another list stores smoothed sentences in which words and phrases are replaced with their synonyms. Since the sentences are ranked by the occurrence of key concepts, the ones with the highest rank are chosen for the summary. Another research [4] builds a benchmark of news articles. Next, it utilizes genetic algorithm to extract the sentences' attributes that affect their classifications. Finally, it applies Naive bayes to classify sentences (summary or Not summary).
Similarly, the effect of attributes is utilized in the structured background knowledge method in [5]. This

[3] https://www.wikipedia.org/
[4] https://www.orpha.net/consor/cgi-bin/index.php

research applies a leave-one-out approach on a predefined set of biomedical attributes to evaluate their importance in the summaries. Then, it collect the attributes that affect the quality of summaries. Finally, the collected attributes are utilized to generate summaries. Last research [6] forms abstractive summary in two steps. First, it prioritizes the dosage regimens via patterns. Each pattern has three properties: duration, dosage and frequency. Second, it applies expert rules on "THERIAQU" dataset to create comments for each pattern to generate the summaries.

In surveys, three research are found. The first research [7] compares some summarization approaches based on summarizing single or multi-documents as shown in Table 1. Some of these approaches utilize (NLG) that is an abbreviation for Natural Language Generation technique.

Table 1. Abstractive Summarization Approaches in [7].

Document Type	Approaches
Single-document	simple structured template and canned generation
Single-document	complex structured template and NLG
Single-document	syntactic representation of sentences and NLG
Single-documents	syntactic representation, ontology annotations and NLG
Single-documents	statistics to rank sentences
Multi-documents	statistics (Support Vector Machine or vector space model)
Multi-documents	inner and outer document relationships

The second research [8] compares ten research papers based on two abstractive summarization approaches: structure based approach and semantic based approach. The comparison includes four factors: techniques, text representation, content selection and summary generation. The first approach depends on the structuring techniques, such as tree, graph, templates, rules, ontology, and lead and body phrase. The second approach depends on the semantic techniques, such as multi-model semantic,information item, semantic graph and reduced semantic graph.

The last research [9] assesses abstractive summarization approaches in the biomedical field. Although it compares different factors, we only include two factors: methodology and corpus as shown in Table 2.

Table 2. Abstractive Summarization Approaches in [9].

Methodologies	Corpora
Support vector machine (SVM) classifies the input question. Then, a conditional random fields model identifies UMLS terms to retrieve relevant texts and similar questions. After that, similarity is measured between the input and the passages (texts and similar questions' answers). The passages	clinical questions database

are summarized based on the terms in the input question	
First, texts are indexed with UML terms. Second, texts are retrieved according to the user's question. Third, they are summarized (sentences are extracted based on the their similarity to the input)	Medline
The user inputs a concept. The system retrieves relevant sentences and creates ranked triples of sentences. The highest ranked triples are organized into a graph. This graph can be utilized for summarization	Medline
MetaMap is utilized to identify texts that are relevant to the user's terms. The retrieved texts are organized using a semantic clustering algorithm. After that, these texts are summarized to include sentences relevant to the input, their titles and research outputs	Medline
First, articles are classified based on the clinical tasks, such as treatment. Second, the articles' outcomes that are similar to the patient's characteristics are retrieved. Third, a shallow syntactic parser is used to match the articles' sentences to pre-defined patterns. Fourth, the matched results are organized in a semantic graph. Last, summaries are generated from that graph	electronic health record and biomedical articles
The graph based summarization method depends on three main technical components and four principles. The technical components are SemRep for extracting semantic predictions, NLP parser for analysing, and UMLS for domain knowledge representation. The principles are connectivity, relevancy, saliency and novelty	Medline
Texts are clustered based on similarity measurement. Features are extracted from each cluster; then, the sentences are ranked based on the features frequency. Top ranked sentences forms the summary	Collection of radiology reports
First, texts are represented in graphs (UMLS terms are the nodes and the relations are the edge). Second, each edge is assigned to specific weight for each node. Third, sentences are clustered. Last, sentences are selected for summarization based on the summation of two factors: the weight of sentences in each cluster and the number of UMLS terms in the sentences	BioMed Central
First, it identifies the outcomes of articles. Second, it utilizes SVM for two purposes: classifying the outcomes into neutral, positive, negative, and extracting the important sentences to generate summarization. The importance of a sentence is determined by its location, its length, including numbers, and it relevancy measurement	Medline abstracts

The UMLS terms are stored in several lists based on their semantic type. Each list is scored based on following factors: the appearance of UMLS terms, the number of exist terms, list length, and 2 the importance of its semantic type. Then, the lists are sorted by the scores, and the important terms are identified from the top lists. Finally, the summary is generated from the sentences that contain the important terms	an article collection of oncology clinical trial
Maximal marginal relevance is applied to rank sentences according to medical terms (unigram and bigrams), terms frequency, terms similarity to the title of the article, novel terms existence, sentence location and length. The summary is generated from the top sentences	medical articles from Internet
It trains a meta-classifier on labelled sentences. The new sentences are ranked according to their predicted class and a set of features: sentence position, length, terms, overlap.	medial news articles
It forms summarization based on hybrid method. In this method, sentences are extracted according to their predicted label, their length, their position, and their relevant to user inquiry	medial news articles
SemRep extracts Semantic relations from sentences. Relations relevant to the user's inquiry are retrieved and ranked for summarization	Medline abstracts
It takes advantage of both domain specific (UMLS) and non-specific (WordNet) knowledge graphs to construct a graph based on documents and the question. The graph's concepts are scored according to their appearance in the question. Then, these scores are utilized to rank sentences. Highest ranked sentences are selected and sorted for summarization	Medline
The sentences of documents are utilized to build a semantic graph. Next, similarity is measured to cluster documents. After that, summarization is applied per cluster to include most common sentences	Medline

2.1.2 Graph-Based Approach

Many research widely apply graph-based approach for summarization. This includes directed, un-directed, and semantic graph. The research [10] constructs a semantic graph for the texts by utilizing a domain specific ontology. Next, the relations in WordNet are applied on the graph to reduce, merge and form the abstractive summary. In the research [11], an algorithm is proposed; it constructs a directed graph given a document and its summary. It starts with extracting the key concepts from the summary, and spans the document sentences to generate new sentences of the key concepts. After that, syntactic rules are applied to combine sentences and create the summary.

In addition, the research [12] generates summaries in two phases. First, it extracts semantic predications from Medline by SemRep; then, it filters the predications graph into a summary graph. Second, it clusters the graph into several labelled summary themes. The optimal theme (the one with the highest similarity measurement to the document) is chosen as the summary. A further research [13] constructs a directed graph for documents. After the graph's paths are scored based on the sentences' redundancy, the sentences are ranked in descending order. The similar sentences are neglected, and the top most ranked sentences are selected for the summary.

Moreover, the research [14] constructs an undirected graph. It divides the sentences into predicate argument structure; then, it measures the similarity among these structures to assign weight for edges. After that, the graph inputs into an algorithm to form the summary. The paper [15] constructs a weighted graph to recognise the top terms within an Indian text. The graph facilitates generating the abstracive summarization. The last research [16] maps documents to UMLS to discover the correlations among the concepts based on the frequency of appeared words. Then, it constructs a graph based on a similarity functions that inputs the correlations. After that, a clustering algorithm is applied to find the themes within the document. The informative and relevant themes are selected to generate the summary.

2.1.3 Neural Network Based Approach

More recent research propose using Neural Network techniques for the abstractive summarization. The research [17] applies Recurrent Neural Network (RNN) with Sequence-toSequence Attentional Model for the encoder and Pointer-Generator Network for the decoder. In addition, two functions are included to to generate better summaries; namely, the maximum likelihood loss and the reinforcement learning objective function. Similar method is proposed in the research [18] with two changes in the encoder and the decoder. It proposes dual encoders model. the first encoder's output is the input for the second encoder. It also inputs an additional auto generated context vector into the decoder. Another research [19] outputs summaries based on the user's query, documents and RNN with a focus on the words overlapping criteria.

In addition, the research [20] applies hybrid Neural Networks. It utilizes LSTM-CNN as encoder and LSTM –RNN as decoder. The input of the encoder is a tree of sentences (Subject phrases, relational phrases, object phrases) and the summarization is the output of the decoder. Moreover, the research [21] incorporates the neural network encoder with BERT. The BERT is utilized to encode the input into context representation vector, and to produce sharper summarizations of the summarizations that are generated by the Transformer-based decoder.

The last research [22] applies Radlex ontology and a dual Neural network. The ontology is for selecting the content of the summary. The content is represented with BERT, and is trained by a bi-LSTM network. The dual encoder Neural network is for generating the summary. While the encoders separately take the found words and the ontological concepts, the decoder takes the encoders' outputs to filter and generate the summaries.

The graph-based research mainly generate abstractive summaries from texts without the utilization of a training set. Referring back to the third objective of this research paper, a training set required to learn how to generate English summaries of Wikipedia articles and Orphadata summaries. The discussed graph-based approach does not utilize any training set. For this reason, neural network-based approach is proposed for this research.

2.2 Semantic Similarity

The semantic similarity methodologies can be utilized to retrieve related/ similar texts into the user enquiry. The literature review shows that these methodologies can be categorized based on the level of similarity into: 1. similarity among concepts, 2. similarity between concepts and sentences, 3. similarity among sentences, 4. similarity between concepts and documents, and 5. similarity among documents. These categorizations are explained next.

2.2.1 Similarity among concepts

There are three main approaches for measuring the semantic similarity among concepts: word embeddings based techniques, graph-based techniques, and hybrid techniques.

2.2.1.1 Word embeddings based techniques

In these techniques, the word embedding techniques are applied first; then, a similarity algorithm is utilized for similarity measurement.

- The first research [23] applies explicit semantic analysis for embedding. Then, Lesklike metric's weighting schema is utilized for similarity measurement.
- The second research [24] applies both GloVe and word2vector for embeddings. Then, it proposes a function to measuring similarity based on both cosine similarity and WordNet based lexical similarity.

2.2.1.2 Graph-based techniques

In graph-based techniques, a graph is parsed or constructed for the concepts. The path between two concepts is utilized to measure their similarity.

- In the research [25], after BabelNet is parsed for the concepts, the shortest path among them is selected for similarity measurement.
- In another research, after the Indian version of WordNet is utilized for extracting a sub graph of the concepts, the path is computed for similarity measurement [26].
- In the last research, after BabelNet is utilized for representing relations among concepts and entities, the conventional closest senses strategy is applied for measuring the similarity score among concepts [27].

2.2.1.3 Hybrid techniques

Hybrid techniques are combination of word embeddings and graph-based techniques.

- In the research [28], vectors (the resulted of multiple modelling resources such as word2vector) are linked with ConceptNet (knowledge graph), so the vectors of the English embedding is propagated via the ConceptNet's multilingual links to the terms in other languages. Then, semantic similarity or relatedness is ranked.
- In additional research, word2vec term representation is enhanced with UMLS (Unified Medical Language System). Then, the minimization of the Euclidean distance of adjacent nodes is used to find the shortest path between the concepts. Last, cosine similarity calculates the similarity measurement [29].

- In another research, the skip-gram neural network model is extended to include the MeSH (Medical Subject Headings) descriptors of the PubMed corpus. In this approach, the concept is represented as a vector by its neighbours in MeSH. After obtaining the pairs of concept/MeSH descriptor, they are input to the neural network by fastText tool. Then, the cosine similarity is measured for the final output [30].
- In further research [31], First, MEDLINE is pre-processed to extract the CUI (concept unique identifier), co-occurrences with each CUI term, and map CUI to metadata to extract similar or related words. Second, Text::NSP, skip gram and word2vector algorithms are used in parallel to reduce dimensionality and, to create vectors of (CUIs, co-occurrences words with CUI, similar or related words to CUI). Third, the outputs of Text::NSP are fed into Matalab for further dimensionality reduction while the outputs of skip gram and word2vector are aggregated (sum and mean operations) for computing the relatedness. This step gives two vectors that represent the paired term via their (CUI, co-occurrences, similar or related words). Finally, the cosine similarity is applied on the vectors of paired terms to compute the relatedness among the paired terms
- In the last research [32], a hybrid semantic relatedness algorithm is proposed to assign a score for the relationship between two biomedical entities. It is based on the co-occurrence and specialized word embedding, and it considers both direct and indirect relationships of the paired entities

2.2.2 Similarity between concepts and sentences

Two research found in this category. They also apply word embeddings or graph based techniques for similarity measurement. The first research [33] measures the similarity between concepts and sentences. It includes 25 target concepts that are restricted to specific senses (adapted from WordNet senses). The senses include (objects, actions, settings, roles, states, and events). After each concept is grouped to a set of 31 sentences, human are asked to define and score the relatedness between the concepts and sentences. Next, the relatedness score is calculated between the concept and sentence using maximum difference scaling and best worst scaling paradigm. Then, the techniques N-gram Latent Semantic Analysis, Word2Vec, UMBC Ebiquity, Word-spotting Baseline (WordSpot), Relatednessspotting Baseline (RelSpot) are evaluated against the human judgments.

The second research [34] measures the similarity between concepts and phrases (two words or more). It proposes three supervised approaches for measuring phrasal semantics:

1. Semantic network model: creating a directed weighted graph based on the relations of words in WordNet and eXtended WordNet. The graph consists of words (the nodes) with labelled edges. Each relation has a weight that is used to measure the closeness between the target word and the words in the phrase via maximum path cost.
2. Distributional similarity model: it uses the Web corpus with window size equal to 3, left and right to collect the words collocate with the words in the target phrase with their frequencies. Then, a vector is created for each word in the target phrase from collocations words. Next, the sum vectorization is calculated for the vectors. Finally, the cosine similarity is calculated between the summed vector and the targeted word vector.
3. Hybrid between the two above.

2.2.3 Similarity Among Sentences

In this category, most of research employ deep learning with other techniques, such as word embedding. The research [35] proposes a model of two components:

1. Convolutional Neural Network (CNN) is used for sentence modelling. It converts the sentence into a concatenation of tokens (temp sequences) to fed them into different types of filters and pools. It has two types of filters: the holistic filters consider entirely each word embedding at each position, and the per-dimension filters consider each dimension of the word embedding. The filter's output is fed into pooling layer to calculate its max, mean and min.
2. Similarity is measured for the local regions of the represented sentence from CNN. The paper proposes an algorithm for comparing similarity in two direction by rows and by columns.

A second research [36] proposes an approach that consists of following steps:

- First, sentences are translated into English through a machine translation, such as Google Translator.
- Second, It builds a universal model to estimate semantic Similarity that combines two methods traditional NLP and deep learning. NLP methods include several features: sequence features, Syntactic Parse Features, alignment features, MT based features, Single sentence feature (BOW, dependency, word-embedding features) and a learning algorithms for regression to assign a score for similarity measurement. The deep learning inputs the word vectors of the pre-trained sentences to measure the semantic.
- Third, the outputs scores of NLP methods and deep learning methods are averaged to get the final semantic relatedness score.

Another research [37] proposes a multi-layers system. It consists of four layers (string similarity, corpus similarity, knowledge similarity, embedding similarity) with multiple algorithms in each layer that can be used separately or jointly. The out put similarity values of these layers are inserted into ML classifiers (e.g. NB or SVM) via Weka for final similarity score.

Last research [38] presents a benchmark of the well known researches conducted in sentences similarity in the period (2012-2017). It compares the papers based on tasks, datasets, approaches, and results. Although the research followed some common methods that includes unifying the language to English, representing sentences in vectors, using cosine similarity for semantic measurement, they are varied in some other techniques that are ordered by performance from highest to lowest as follows (only the top four research are listed):

1. The first one is ECNU; it uses three feature engineered models: Random Forest (RF), Gradient Boosting (GB) and XGBoost (XGB) regression methods. The features are based on (n-gram overlap; edit distance; longest common prefix/suffix/substring; tree kernels; word alignments; summarization and MT evaluation metrics; and kernel similarity of bags of-words, bags-of-dependencies and pooled word embeddings). These models are employed with four deep learning methods (averaged word embeddings, projected word embeddings, a deep averaging network (DAN) or LSTM). Each network has additional layers to feed it with the result of three operations (multiplication, concatenation, and subtraction) that are executed on the vectors of the paired sentences to produce similarity scores. At the end, the average similarity score is calculated.
2. The second one is BIT. It uses sentence information content (IC) that is fed by WordNet and BNC word

frequencies. It presents different experiments. The best one is combining IC with cosine similarity. The cosine is calculated based on the sum of the IDF weights for the vectors.

3. The third one is HCTI. It uses a model that is similar to the convolutional Deep Structured Semantic Model (CDSSM). It generated the vectors using twin convolutional neural networks (CNNs); then, it compares the similarity using elementwise difference and semantic similarity. After that, the similarity values are input into extra layers to assign labels to the similarity.
4. The fourth one is MITRE that is similar to ECNU; it merges feature-engineering methods (alignment similarity; TakeLab STS; string similarity measures such as matching n-grams, summarization and MT metrics (BLEU, WER, PER, ROUGE)) with deep learning methods (RNN and recurrent convolutional neural networks (RCNN) over word alignments; and a BiLSTM).

2.2.4 Similarity between concepts and document

Only one research is presented in this category. The research proposes [39] computational framework to input several news articles (more than 500,000 articles), and to extract the semantic relatedness among the mentioned cities in a specific topic. The framework quantifies the semantic relatedness between two cities as the number of news articles that contain the two cities and discuss the topic.

2.2.5 Similarity among documents

This category also presents only one research. The research [40] measures similarity based on synonyms among words not on co-occurrence and usage. It starts pre-processing the documents with the NLP techniques to extract the keywords and their stems. Next, it applies feature extraction to re-arrange words based on their POS, and synonym extraction to retrieve synonym from WordNet based on keywords' stems. After that, the SQL semantic DB is used to assign weights to the keywords and synonyms of each document. Then, the weights of similar words are summed, and the percentage is calculated as the similarity measurement. Finally, both keywords and synonyms are visualized with Graphviz library.

Since one objective of this research aims at finding the similar concepts that are related to the user inquiry in the summaries corpus, an approach of the similarity among concepts approaches can be chosen to fulfill this objective.

3. Data Considerations

This section discusses two types of data: data for summarization, and data for translation.

3.1 Data for English Summaries Corpus

This corpus consists of texts that are taken from Orphadata [5]. Orphadata aims at providing "the scientific community with comprehensive, quality data sets related to rare diseases from the Orphanet knowledge base". It offers several reusable data sets that can be utilized for "research, educational and informational purposes only".

These data sets are XML based files to facilitate retrieving disease titles, summaries and symptoms; these

[5] https://orphadata.org

are the three main components of the corpus as will be shown next. We propose developing a Python tool to automate building this corpus. This section only discusses two activities of building the corpus: metadata description and text selection.

3.1.1 Metadata Description

The corpus requires metadata description. It is included in XML based file that contains the following information about data: data sources, creator, purpose, date, disease title, unique disease identifier, summary, symptoms. This corpus is only available in English Language.

3.1.2 Text Selection

This part presents the XML files that are required to develop the corpus. It also lists the three main components of the corpus, and the files' attributes names where data is originally stored. All these files are open access and under the Commons Attribution 4.0 International (CC BY 4.0) Licence.

- The disease titles are taken from the file: "ORPHAclassification_156_Rare_genetic_disease_e00 Specifically, the data is taken from the child tag < Name > of the parent tag < disorder id >. The following code has been excerpted from the named file for clarification.

```
<Disorder id="20039">
<OrphaNumber>263297</OrphaNumber>
<Name lang="en"> Glycogen storage disease with severe cardiomyopathy due to glycogenin deficiency
….
</Name>
</Disorder>
```

- The summaries are taken from Orphadata [6]. They are stored in the XML based file: "ORPHAnomenclature_en00; specifically in the tag < contents >.
- The symptoms (diagnostic criteria) are taken from Orphadata [7]. It is stored in "en_product400, specifically in the < HPOTerm > tag. The next code illustrates the associated symptoms for a specific disease and its frequency.

```
<HPODisorderAssociation id="20039">
<HPO id="1073">
<HPOId>HP:0100490</HPOId>
<HPOTerm>Camptodactyly of finger</ HPOTerm>
</HPO>
<HPOFrequency id="28426">
<Name lang="en">Occasional (29-5%)</ Name>
</HPOFrequency>
<DiagnosticCriteria/>
</HPODisorderAssociation>
```

[6] http://www.orphadata.org/cgi-bin/ORPHAnomenclature.html

[7] http://www.orphadata.org/cgi-bin/index.php

It is noticeable that the tag < disorder_id > has the same value “20039” across the three codes even if the tag name is slightly different. It is a unique disease identifier that shared among the files. Therefore, it can be utilized to retrieve relevant data from these three files. After collecting data from the files, the corpus for the English summaries should be created as follows:

```
<Disorder id="20039" lang="en">
    <Name> Glycogen storage disease with severe cardiomyopathy due to glycogenin deficiency
    </Name>
     <Summary> Glycogen storage disease type 15 is an extremely rare genetic glycogen storage disease
    reported in one patient to date. Clinical signs included muscle weakness, cardiac arrhythmia associated
    with accumulation of abnormal storage material in the heart and glycogen depletion in skeletal muscle.
    </Summary>
    <SymptomsList>
    <SLId id=01>
    <SLName>
    Camptodactyly of finger
    </SLName>
    <SLFrequency>
    Occasional (29-5%)
     </SLFrequency>
    </SLId>
    </SymptomsList>
</Disorder>
```

3.2 Data for Arabic-English Translation

Searching the literature for medical Arabic-English corpora shows no results. However, non medical (Arabic-English) corpora are available for the translation purpose. For instance, the research [41–43] construct a corpora of general un-annotated data while the research [44] builds a corpora of political meta-annotated data.

In this research, Wikipedia has been investigated to check the possibility of utilizing its medical texts to build bilingual parallel corpora. Wikipedia [8] is knowledge base website that depends on community contribution to modify its contents. It has articles that cover different topics, such as medicine and technology. The most dominant language is for English articles but some articles are available in other languages as well. It has more than 125 languages.

Wikipedia has been used as knowledge base in many previous research. In the research [23], Wikipedia and WordSimilarity-353 dataset are utilized to create six cross-lingual dataset to facilitate finding similar related words of different languages. Lesk-like metric measures the similarity and relatedness between individual terms in different languages [23].

In addition, Wikipedia articles are utilized to extend UMLS (Unified Medical Language System) model via

[8] https://en.wikipedia.org/wiki/Wiki

extracting derived concepts that are related to UMLS concepts using cosine similarity function. After extracting the derived concepts from Wikipedia, the relationships between these derived concepts and UMLS's concepts are determined as parent/child or boarder/narrower relationships; next, the concepts' definitions are retrieved from Wikipedia. Finally, an extension of UMLS model is presented by crating a sub graph of the derived concepts, their definitions, and their relationships [45].

Moreover in [46], the Wikipedia inter-language links are incorporated to unify the languages of a text without the need of machine translation process. This research proposes a method for semantic relatedness that inputs a text in specific language and output the similar texts in another language; it is extending Bayes theorem (ENB) for computing similarity among the generated vectors. The links incorporation could be achieved by two approaches:

- SimpleMap: it takes out the input's terms and map them to equivalent English terms. Any term that does not have inter-language link is not mapped.
- ProbMap: it considers two situations when it maps the input's terms to the English terms. If input terms has inter-language link, it directly maps terms, and if the input's term does not have an inter- language link, it's mapped to similar term. The two methods output vectors of English texts. ENB uses these vectors to compute semantic similarity among the texts

Many other exist research use Wikipedia as knowledge base [47], [48] and [49]. The content of Wikipedia corpus can be obtained via a Python tool, similar to [50]. It can fully downloaded or only the related pages can be extracted via the titles search[9]. Next, some of the activities, that are required for building parallel corpora, are discussed.

3.2.1 Metadata Descriptions

This research proposes creating XML based files that includes the following information about the data: data source, date creator, article title , unique disease identifier, English text, Arabic text. The article title is the disease title, and the unique disease identifier that is utilized to uniquely identify each disease.

3.2.2 Text Selection

3.2.2.1 Articles Selection

Wikipedia's articles are retrieved if they have titles equal to/similar to the disease titles that are taken from[10]. The titles are stored in the XML based file:

"ORPHAclassification_156_Rare_genetic_disease_en00 within the child tag < Name >. The unique disease identifier is also taken from the same file; it exists within the child tag < Disorderid = ".....” >. Both English and Arabic article versions are retrieved from the Wikipedia.

3.2.2.2 Wikipedia's Sections Selection

Wikipedia has some very long articles; therefore, we should determine what Wikipedia's sections are selected of the articles in interest. In table 3, w manually compared the texts from three different data sources: Wikipedia, Orphadata and Medline for some diseases. These include Pseudoxanthoma Elasticum,

[9] https://medium.com/@Alexander_H/scraping-wikipedia-with-python-8000fc9c9e6c

[10] http://www.orphadata.org/cgi-bin/rare_free.html

CARASIL, HANAC syndrome, Turner syndrome. Although we have compared more than four diseases, these diseases are randomly selected for clarification. The goal of this comparison is to determine the Wikipedia's sections that are required to cover the medical terminologies in the Orphadata summaries. This comparison also shows that Medline may do not have summaries for all mentioned diseases; therefore, Orphadata is selected for the English summaries in the previous section. The texts of the above mentioned diseases are obtained as follows:

1. Python[11] is utilized to retrieve Wikipedia's texts. It has some built-in functions to retrieve the texts of Wikipedia's summary and sections.
2. Orphdata texts are taken from the XML based file: "ORPHAnomenclatureen00; specifically, it is the value of the tag $<$ content $>$.
3. Medline texts are taken from the XML based file: "mplustopics2020 − 05 − 26.txt00; specifically, it is the value of the tag $<$ full − summary $>$.

Four criteria are considered during the comparison process: the accuracy of text retrieval, the length of texts, the format of texts, and the terminologies. These are the findings of the comparison:

1. Python code has failed to retrieve the appropriate Wikipedia article for the second disease "CARASIL".
2. Despite the fact that the summaries of Wikipedia articles might be long, at least two Wikipedia sections, namely history, and sign and symptoms, are required to cover the medical terminologies that are exist in the summary.
3. Both Wikipedia and Orphdata have the same text format (a set of sentences) but Medline sometimes contains "questions and answers" format besides the textual one.
4. Wikipedia and Medline many not have information about some Orphadata's diseases. If Wikipedia does not have an article about the disease, the disease is neglected.
5. Texts of Orphdata are always shorter than Medline.

Table 3. Texts of Genetic Diseases from Various Sources.

Wikipedia	Orphadata	Medline
Pseudoxanthoma elasticum (PXE) is a genetic disease that causes mineralization of elastic fibers in some tissues. The most common problems arise in the skin and eyes, and later in blood vessels in the form of premature atherosclerosis. PXE is caused by autosomal recessive mutations in the ABCC6 gene on the short arm of chromosome 16	Pseudoxanthoma elasticum (PXE) is an inherited connective tissue disorder characterized by progressive calcification and fragmentation of elastic fibers in the skin, retina, and arterial walls	Only disease's title

[11] https://pypi.org/project/wikipedia/

The caracal (Caracal caracal) is a medium-sized wild cat native to Africa, the Middle East, Central Asia, and India. It is characterised by a robust build, long legs, a short face, long tufted ears, and long canine teeth, etc.	CARASIL is a hereditary cerebral small vessel disease characterized by earlyonset gait disturbances, premature scalp alopecia, ischemic stroke, acute mid to lower back pain and progressive cognitive disturbances leading to severe dementia	disease does not exist
disease does not exist	A rare multisystemic disease characterized by small-vessel brain disease, cerebral aneurysm, and extracerebral findings involving the kidney, muscle, and small vessels of the eye	disease does not exist
Turner syndrome (TS), also known 45,X, or 45,X0, is a genetic condition in which a female is partly or completely missing an X chromosome. Signs and symptoms vary among those affected. Often, a short and webbed neck, low-set ears, etc.	Turner syndrome is a chromosomal disorder associated with the complete or partial absence of an X chromosome	Turner syndrome is a genetic disorder that affects a girl's development. The cause is a missing or incomplete X chromosome.

3.2.3 Text Segmentation and Alignment

Texts are segmented by sections/paragraphs that are aligned with their equivalent Arabic translations. Since texts have several paragraphs, the unique disease identifier connects the paragraphs that belong to the same article. Since this corpora is built for the Arabic-English translation task, the segmentation speeds up the translation. This is because it facilitates working directly on paragraphs (set of sentences and their translations) instead of working on long document. This minimizes the number of sentences that will be translated at a specific time.

3.2.3 Annotations

After the text segmentation activity, UMLS[12] can be applied to annotate the medical concepts within the Wikipedia's texts. These annotations contribute to improve disease-text relevancy via fostering the

[12] https://www.nlm.nih.gov/research/umls/index.html

existence of medical terminologies within the text. Specifically, the UMLS attribute <SemanticType>[13] can assist on deciding the annotated value for the UMLS concepts after the retrieval. This attribute has several values. This research consider only two values that are important for the main components of the English summaries corpus:

- Disease or Syndrome: the Disease value annotates the disease titles within Wiki texts.
- Sign or Symptom: the Symptom value annotates the disease symptoms within Wiki texts.

The following list illustrates the UMLS files and the required attributes for the annotations:

1. "MRCONSO.RRF.aa" and "MRCONSO.RRF.ab" are two files that include several attributes. The important ones are the concept unique identifiers, their language and the concepts.
2. "MRSTY.RRF" is a file that includes two attributes: the concept unique identifiers and their semantic type identifiers.
3. "SRDEF.RRF" is a files that includes many attributes. The important ones are the semantic type identifiers and their values.

Table 4 clarifies a sample row of these UMLS files. The bold texts are the possible values of the listed attributes. For instance, the concept unique identifier is C0002895, its language is ENG, its value (the concept itself) is Anemia, Sickle Cell, and its semantic type value is Disease or Syndrome. While the concept unique identifier connects the two files: "MRCONSO.RRF" and "MRSTY.RRF", the semantic type identifier connects the two files "MRSTY.RRF" and "SRDEF.RRF". Using these two attributes facilitates retrieving the semantic type for a specific UMLS concept.

Table 4. Contents of Required UMLS Files for Annotations.

<table>
<tr><th>File Names</th><th>Sample Row</th></tr>
<tr><td>"MRCONSO.RRF.aa"
"MRCONSO.RRF.ab"</td><td>C0002895 | ENG | <i>P</i> | <i>L</i>0002895 | <i>PF</i> | <i>S</i>0013783 | <i>N</i> | <i>A</i>0023707 || <i>M</i>0001140 | <i>D</i>000755 | <i>MSH</i> | <i>MH</i> | <i>D</i>000755 | Anemia, Sickle Cell | 0 | <i>N</i> | 256 |</td></tr>
<tr><td>"MRSTY.RRF"</td><td>C0002895 | T047 | | ||||</td></tr>
<tr><td>"SRDEF.RRF"</td><td><i>STY</i> | T047 | Disease or Syndrome | <i>B</i>2.2.1.2.1 |A condition which alters or interferes with a normal process, state, or activity of an organism. It is usually characterized by the abnormal functioning of one or more of the host's systems, parts, or organs. Included here is a complex of symptoms descriptive of a disorder.|| Any specific disease or syndrome that is modified by such modifiers as "acute", "prolonged", etc. will also be assigned to this type. If an anatomic abnormality has a pathologic manifestation, then it will be given this type as well as a type from the "Anatomical Abnormality" hierarchy, e.g., "Diabetic Cataract" will be double-typed for this reason.|| <i>dsyn</i> ||</td></tr>
</table>

[13] https://uts.nlm.nih.gov/metathesaurus.html

A Python tool can be developed to annotate Wiki texts via employing the above files. This tool not only annotates the Wiki texts but also creates a bilingual medical dictionary (English-Arabic) of the exist UMLS concepts in the Wiki texts through the following steps:

1. It parses Wikipedia texts to extract UMLS terminologies using NLP techniques.
2. It checks if a terminology does not exist in the dictionary.
3. If a terminology does not exist in the dictionary but exists in UMLS, it searches Wiki texts to locate it. If it is found, it is annotated, and its Arabic translation is annotated, as well. A method of how to identify the Arabic version of a terminology should be investigated. One method could be using a machine translation technique to automatically translate these annotated English terminologies.
4. Both the English terminology and its Arabic translation might be stored in the dictionary with their semantic type.

This dictionary facilitates recognizing medical terminologies and mapping them to their Arabic versions which might be useful for summarization and translation tasks.

4. Research Methodology

This section discusses the implementation requirements of the last four objectives in this research: learning English abstractive summaries, translating these summaries into Arabic to develop Arabic equivalent version of the English summaries corpus, semantic similarity retrieval, and system evaluation. We propose using RNN for both summarization and translation. The next sections explain the proposed methodology with more details.

4.1 Proposed System

This research proposes applying a supervised learning methodology with Recurrent Neural Network (RNN). This section, first, presents the problem dimensions; second, it states the system's inputs; third, it discusses the system architecture from different aspects, such as cells components, hyperparameters and possible used framework.

4.1.1 Problem Dimensions

This includes the inputs, outputs, their characteristics and RNN tasks.

1. Inputs: a sequence of selected texts from Wikipedia or Orphadata. This is further discussed in the next section.
2. Outputs: a sequence of texts that represents two kind of outputs : internal (intermediate outputs) and external (final outputs)

- First internal outputs: this includes the English summaries of Orphadata and Wiki texts; the first RNN produces the Wiki summaries by learning Orphadata's summaries.
- Second internal outputs: this includes the Arabic translated version of Orphadata and wiki summaries; the second RNN produces the Arabic version of the first outputs by learning Arabic-English Wiki texts.
- Final outputs: The equivalent Arabic version of data in the English summaries corpus that consists of disease title, summary, and symptoms for each disease.

3. In/out put characteristics: Each in/out put may have a varied length of sequence; in addition, they might be of either Arabic or English language.
4. RNN tasks: RNN learns how to; first generate summaries for English Wikipedia texts, and second, translate the English summaries into Arabic.

4.1.2 Inputs

This system has two main data: training sets and a dictionary. The training sets are created from the data that is already discussed earlier in Section 3. There are two training sets. The first training set is utilized for the summarization task; it has the English Wiki articles as the inputs and their Orphadata summaries as the outputs. The second training set is utilized for the translation task; it has the English Wiki articles as the input and their Arabic versions as the output. The texts are extracted from the training sets, and preprocessed to create the dictionary (a bilingual word embedding representations) following these steps:

1. Texts are lowered cased and tokenized. Tokenization should recognize complex terms, such as medical terms that consist of more than one word. This research only consider identifying the annotated medical terms within the texts.
2. The available Arabic English semantic networks might be utilised as an additional data resource for the dictionary. This could contribute in solving the NLP problem: "Out-Of-Vocablury"(OOV) error.
3. The English vocabulary and it's equivalent Arabic are assigned to the same index to facilitate the retrieval of vocabulary in either languages.
4. Avoid repetition or duplicate terms.
5. Word embeddings techniques, such as FastText [51] or Word2vec [52], are utilized to obtain terms embedding representations.

Since the dictionary contains bilingual vocabulary from Wiki and only English vocabulary from Orphadata, this may cause running into "Out-Of-Vocablury" error for some Arabic vocabulary. Therefore, the dictionary may take advantage of the available semantic networks, namely BabelNet [53] or ConceptNet [54], to solve this error.

Table 5 compares BabelNet and ConceptNet semantic networks from different aspects in order to decide which one is more useful for this research. Although these both semantic networks are compatible with Python, this research proposes utilizing BabelNet because it is fully integrated with Wikipedia, and it offers more bilingual vocabulary. Figure 1 shows the possible sources of the terms in the dictionary.

Table 5. Semantic Networks Comparison.

Semantic Network	BabelNet	ConceptNet
Website	https://babelnet.org/	https://conceptnet.io/

Definition	"BabelNet is both a multilingual encyclopedic dictionary, with lexicographic and encyclopedic coverage of terms, and a semantic network which connects concepts and named entities in a very large network of semantic relations, made up of about 16 million entries, called Babel synsets. Each Babel synset represents a given meaning and contains all the synonyms which express that meaning in a range of different languages."	"ConceptNet is a multilingual knowledge base, representing words and phrases that people use and the common-sense relationships between them."
Availability	freely available under the Creative Commons Attribution: https://babelnet.org/ license	licensed under a Creative Commons Attribution-ShareAlike 4.0 International License: https://creativecommons. org/licenses/by-sa/4.0/
Vocabulary	English:3229307- Arabic:2018295	English:1803873 - Arabic:134311
Others	Automatically integrated to WordNet and Wikipedia articles and many other resources	Connected to a subset of DBPedia (extracts knowledge from Wikipedia articles), and much of its knowledge is from Wiktionary (a free multilingual dictionary)

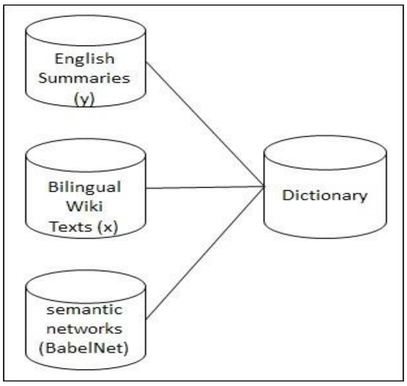


Figure 1. The Sources of Vocabulary in Dictionary.

4.1.3 System Architecture

This part explains the proposed architectures that are based on Bi-directional attentional sequence to sequence Recurrent Neural Network (RNN), and its main features. RNN [55] is basically consists of several layers of functions that are connected. It makes predictions based on learning how to represent data (inputs) and how to transform inputs into outputs. This transformation must consider several computational operations. Such operations could be distributed among three main steps [55]:

1. Forward propagation: this step aims at learning the behaviour of a linear function 4.1 on the inputs (x), weights (w), and biases (b) to produce the corresponding outputs. The step involves the linear function, follows by an activation function [56], such as Equation 4.2 to predict the output(y'). At the beginning, both weights and biases are randomly initiated. At the end, a loss function evaluates the predictions of the linear function for the given inputs to learn how to improve the parameters (weights and biases) to give better prediction. It measures the loss of making a prediction of output y' while the true output is y.

$$f(x) = x * w + b \tag{4.1}$$

$$a = \tanh f(x) \tag{4.2}$$

2. Backward propagation: this step calculates the derivatives (gradients for parameters) to improve the predictions via minimizing the loss.
3. Gradient descent: this step updates the parameters to improve the learning, such as the weights that are involved in the calculations of the linear function.

These three steps are included in any basic RNN. However, since this papers proposes using RNN for NLP tasks, namely summarization and translation. RNN should have some additional features that are RNN memory cell components. For this purpose, we next presents the overall system architecture; then, we discuss these features. Figure 2 clarifies the two steps involved in the system architecture that is adapted from the paper [57].

In the first step, the system learns how to generate the English summaries. The encoder takes the English embedding representations of Wiki texts; and the decoder outputs the predicted English summaries. The RNN features are shown in Figure 3 that is adapted from the paper [58]. It consists of two-level encoders and one decoder; they are connected via two attention models. The two-level encoders encode the Wiki articles (the input) within two levels: the section and the words levels. The decoder is to predict the abstractive summarization for the input. The input is taken from the parallel corpora where one Wiki article is segmented into several sections. Therefore, the tag < disorder_id > is utilized to concatenate these sections into a single section that has several sentences. The section attention model is utilized to encode the section discourse, and the words attention model is utilized to encode the sentences within the section.

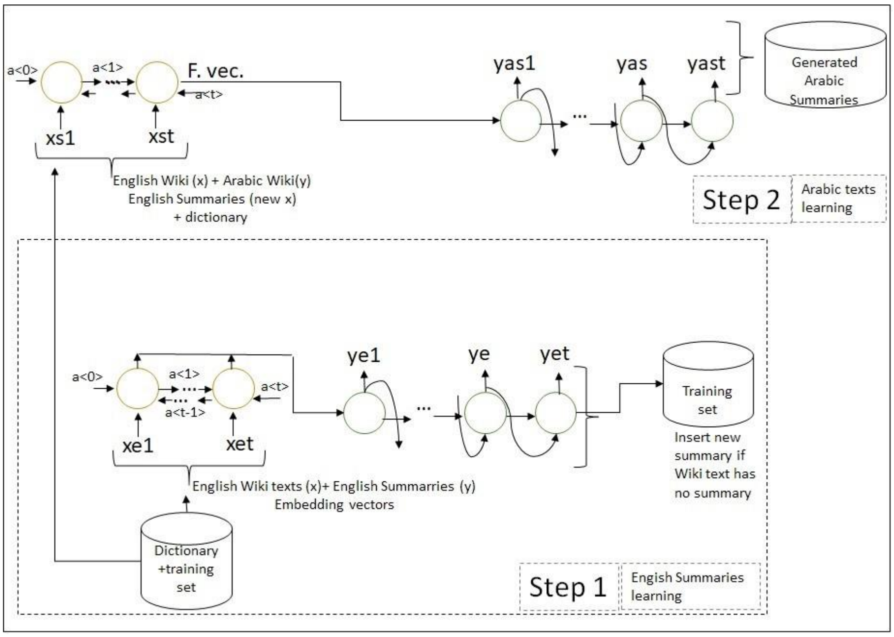

a<0>
a<1>
F. vec.
a<t>
xs1
xst
English Wiki (x) + Arabic Wiki(y)
English Summaries (new x)
+ dictionary
yas1
yas
yast
Generated Arabic Summaries
Step 2
Arabic texts learning
a<0>
a<1>
a<t-1>
a<t>
xe1
xet
ye1
ye
yet
Training set
Insert new summary if Wiki text has no summary
English Wiki texts (x)+ English Summarries (y)
Embedding vectors
Dictionary +training set
Step 1
Engish Summaries learning

Figure 2. System Architecture.

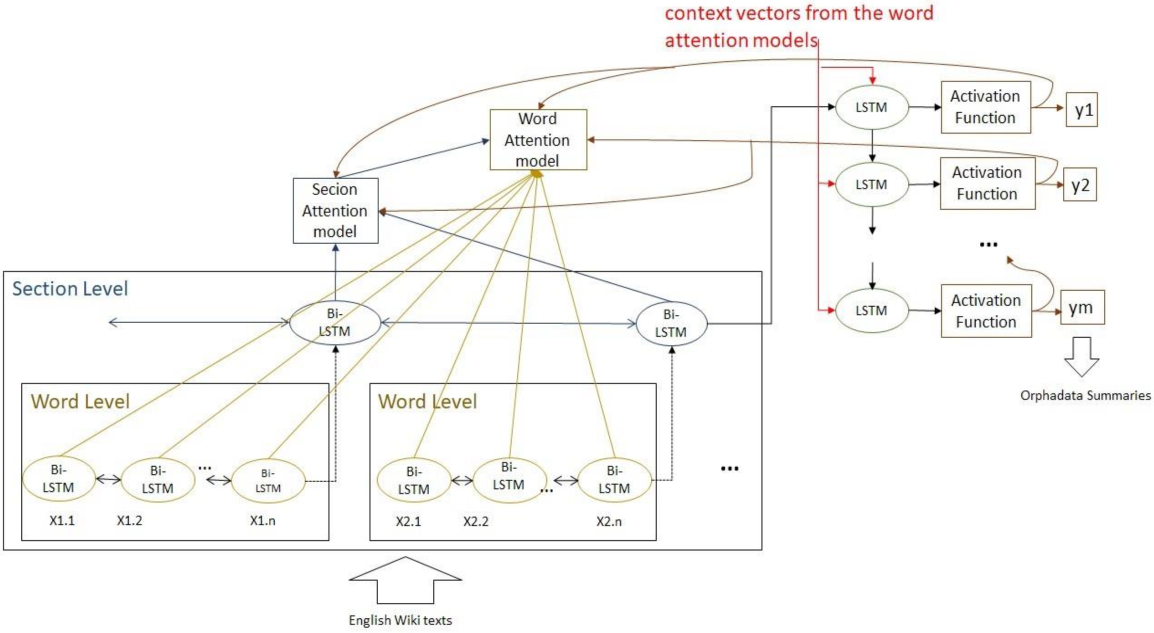


Figure 3. RNN Memory Cell Components for the Summarization Task.

In the second step, the system learns English-Arabic translation. It takes the bilingual embeddings representation of the English texts as the input of the encoder. The decoder outputs the predicted Arabic version texts. Figure 4 shows the RNN features that are involved in this step. They are adapted from the paper [59]. It consists of one encoder and one decoder that are connected to an attention model. The encoder is to encode the English sentence that is taken from the Wiki articles, and the decoder is to predict its translated version in Arabic. This step utilizes one encoder because the translation task differs from the summarization task. It does not require collecting the several sections that belong to the same Wiki article to train the system. The system is trained on each section separately.

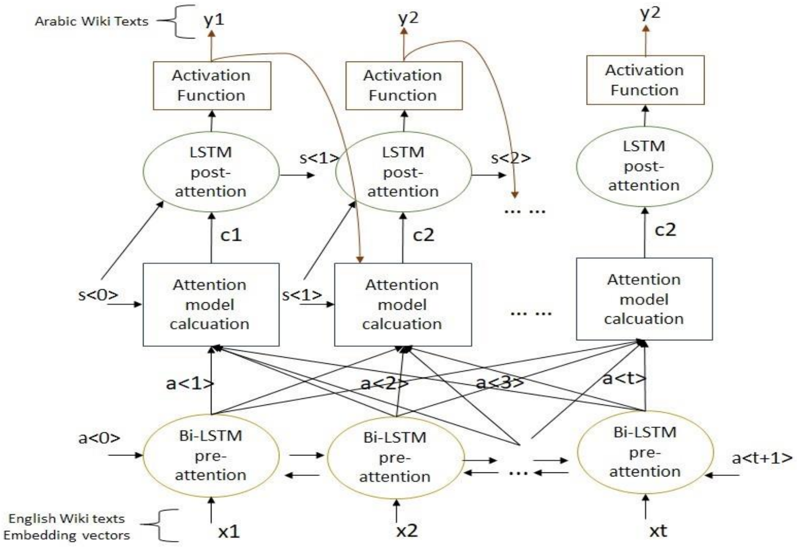


Figure 4. RNN Memory Cell Components for the Translation Task.

Next, the RNN features are clarified. This includes the number of layers, the encoder-decoder model, the bi-directional model, the long short term memory model, and the attention model.

1. Layers:

Deep neural network refers to neural networks with several layers [55]. These layers may consist of the input layer (encoder), one hidden layer, and the output layer (decoder) that are associated with the weights (W), biases (b) and the activation functions. In this research, the number of cells in the encoder is determined by the number of words per input (X). Similarly, the number of cells in the decoder is determined by the number of words per output (Y). However, the number of the hidden layers and the number of cells in the hidden layer are determined by the developer. RNN is very complex; therefore, many systems adapt less than or equal to three layers.

2. Encoder-Decoder Model:

The encoder-decoder model [57] brings two main advantages to the neural networks. First, it is a solution for reading inputs and producing outputs that are sequences of variable length. Since each input and each output may have variable length (different words count), the length of the inputs must be unified. Similarly for the length of the outputs. The unification process is achieved by, first, choosing the sentence that has the maximum length (no of words) among the texts in the training set, e.g. the inputs (X). Second, any sentence that has less length should be padded with zeros. Second, it facilitates creating hybrid neural network as the one that is already stated in Chapter 2, namely, the research [20] applies Convolutional Neural Network (CNN) as the encoder and Recurrent Neural Network (RNN) as the decoder.

3. Bi-directional Model:

Bi-directional [60] means that the neutral network has two states: forward and backward state. Both states are independent, and calculate a function called the activation function in reverse order. The forward state starts from the beginning of the sentence to its end while the backward state works with the opposite direction.

4. Long Short Term Memory Model:

The long short term memory model (LSTM) [61–63] solves the problem of gradient descent vanishing that happens in the backward propagation through time, and affects the weights of the inputs. LSTM has several additional calculations inside each memory cell at an activation time t that include followings:

a. Candidate memory cell value (update_cell_value). It stores a new memory cell value that is utilized to update the current memory cell value.

$$Ucell\_value_t = \tanh(W_c[a^{t-1}, x^t] + b_c) \qquad (4.3)$$

b. Three gates control the inputs to the memory cell. They are closed, if their values are around zeros.

$$\text{Forget gate: } F\_gate = \sigma(W_f[a_{t-1}, x_t] + b_f) \text{ (b)} \qquad (4.4)$$

$$\text{Update gate: } U\_gate = \sigma(W_u[a_{t-1}, x_t] + b_u) \text{ (c)} \qquad (4.5)$$

$$\text{Output gate: } O\_gate = \sigma(W_o[a_{t-1}, x_t] + b_o) \qquad (4.6)$$

c. Current memory cell value: it is also called the cell internal state. It is a local value for each memory cell. cell_valuet = U_gate × Ucell_valuet + F_gate × cell_valuet−1 (4.7)

d. Activation function:

$$a_t = O\_gate \times \tanh(cell\_value_t)$$

5. Attention Model:

The attention model [64, 65] facilitates working with long inputs sequence (long sentences) in the encoder although it has high run time complexity (quadratic). This time complexity is computed by multiplying the length of the input (x) by the length of the output (y), so run time = length(x) × length(y).

This means that attention model takes longer time when the input sentences get longer. The employment of CNN as the encoder instead RNN could be a solution. Reducing the number of parameters is one advantage of CNN over RNN. However, the attempts for solving this issue must be further investigated since our system proposes utilising three sections of Wiki texts that consist of several paragraphs for this research objectives.

Despite this drawback, the attention model improves the system's results because it allows RNN to work on subsets of sentences instead of working with only single long input vector of sentences. It encodes the inputs into several vectors; then, it calculates the probabilities of the subset vectors during the decoding process to select the words combination that has the highest probability among the other combinations at a specific time step t. It applies the following calculations to calculate the probability of a true word y_t with other words combination.

$$p(y_t|y_1,\ldots,y_{t-1},x) = g(y_{t-1},s_t,c_t) \qquad (4.9)$$

where x is the input sentence, and g is non linear function that depends on two computed variables: the variable s_t is a hidden state, and the variable c_t is a context vector. The variable s starts with the initialization vector $< s_0 >$ at time step $t = 0$; it is initialized with zeros. When $t > 0$, s_t is computed as follows:

$$st = f(st-1, yt-1, ct) \qquad (4.10)$$

The variable c_t is computed as follows:

$$c_t = \sum_{j=1}^{len(x)} \alpha_{tj} a_j \qquad (4.11)$$

The αtj is a weight that shows the amount of attention yt should pay to each cell (word) activation aj. Since it is a wight, it is summation should be equal to 1: $\sum_{j=1}^{len(x)} \alpha_{tj} = 1$. It is computed by this equation:

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{n=1}^{len(x)} \exp(e_{tn})} \quad (4.12)$$

where $e_{tj} = uf(s_{t-1}, a_j)$ and $e_{tn} = uf(s_{t-1}, a_n)$. The function uf is an unknown function that is learned by the RNN model. Although there are two attention models for the summarization task, the same equation 4.11 is applied.

6. Other Features

Some additional features that should be determined for the RNN model are:

- The possibility of using Beam search [66] to improve the translation task.
- The selection of the activation functions [67] for layers.
- The selection of a function optimization to improve the model's accuracy and performance.
- The weights initialization process (assigning zeros or random numbers).
- The selection of a learning rate value.

4.2 System Software

The paper [68] compares the available Deep Learning frameworks and libraries via stating the strong and the weak points of each model. This research proposes using Python for the implementation. Python is one of the most famous general purpose programming languages [69]. It has several features [68, 70] that make it suitable for implementing Natural Text Processing (NLP) and deep leaning (DL) techniques. One of these features is its library modules which will be discussed next.

4.2.1 NLTK

NLTK [71] is an abbreviation for Natural Language Toolkit. It offers several libraries that facilitates the followings:

- Accessibility: it facilitates accessing different textual resources, such as corpora and lexicons.
- Pre-processing: it offers different functions for texts analysis, such as tokenization, parsing, word tagging, and stemming.
- Classification: it allows classifying texts and documents.

For the purpose of this research, only the first two types of libraries are important. The first type assists on accessing Wikipedia and Orphadata to create the training sets. The second type assists on annotating the parallel corpora and extracting the vocabulary.

4.2.2 Gensim

Gensim [72] is utilized to convert any text (set of words) into semantic vectors. These vectors are a way to project the words onto a mathematical space. These vectors facilitate applying machine learning algorithms on the texts to perform a specific task, such as prediction [73]. Gensim offers several models to obtain these vectors, such as Word2vec and FastText models. In addition, it offers several techniques, such as similarity measurement and text summarization [52]. In this research, Gensim is utilized for two reasons:

- Word embeddings: it provides the words' vectors (embeddings) that are the inputs (X) into the RNN.

- Similarity measurement: it Measures the similarity between the user inquiry and the data as stated in the next section 4.3.

4.2.3 Numpy and SciPy

Both Numpy and SciPy [74] are used for scientific computing in Python. Numpy extends Python with fast multidimensional array-processing capabilities. It provides several mathematical related functionalities, some are:

- Array initialization, such as creating array of zeros.
- Element-wise computation with array.
- Mathematical operations among arrays.
- Linear algebra operations and random number generation.

SciPy addresses some standard problem domains in scientific computing. For instance, it includes packages for matrix decomposition, function optimization, and solving differential equations. These two libraries assist with the calculations that are required in RNN, such as the calculations in the forward propagation.

4.2.4 Keras

Keras [68] provides a connection between Python and Deep Learning tools, e.g., TensorFlow. Some of its features are the followings:

- Any Deep Learning model is considered as a sequence of fully configurable independent modules that can be combined together. For instance, inputs initialization, outputs, cost functions, neural network layers, optimization function, and activation functions are all independent modules that are offered by Keras, and called separately. They are combined to make new models.
- New modules are easily added, such as adding new neural network layer to the current architecture.
- Although the first feature facilitates the implementation, it limits the flexibility of Keras. Therefore, it is not optimal for checking new architecture. Another drawback of Keras is related to its efficiency. Many benchmarks have stated that TensorFlow backend Keras is not 100% efficiently working in Multi-GPU platform.

4.3 Semantic Similarity Retrieval

This research only considers similarity among concepts (single terms). The similarity is measured between the concepts that are inserted by the user and the concepts that exist in the summaries corpora. The user is limited by a maximum of three words of one language per inquiry. The concepts may include medical terms, symptoms, or disease titles in either English or Arabic. In case the user inputs a phrase of three terms, the system separates it into a set of terms. Then, it searches the English-Arabic summaries corpora to retrieve data that contains exact match or similar terms of the inputs' language. A threshold can be set to determine the minimum value of the similarity that is accepted among the terms.

The exact match terms can be found using Python string built-in functions while the similarity among terms can be measured using semantic similarity methodologies. Since bilingual embedding dictionary is already created for RNN, it can be utilized with any semantic similarity measurement methodologies, such as the "Cosine Similarity" that is in FastText [51] or Word2vec [52]. These methodologies assist finding similar/related terms to the user's concepts in English or Arabic.

4.4 System Evaluation

Translation and summarization should be evaluated. The accuracy of these two tasks can be evaluated using Bleu Score [75] but unfortunately there are no gold standards testing sets for the proposed training sets, namely English-Arabic summaries in medicine. The paper [76] discusses many available methods for evaluating the translation task; these methods might be applicable for evaluating the summarization task, as well. They could be categorized into the followings:

1. Qualitative method: the system's users measure the quality of the system's outputs by assigning an accuracy value for a set of randomly selected outputs. The value could have one of five possible scale values (Not translated, mistranslated, poor, fair, good).
2. BLEU Score method: it is utilized in combination of system outputs with one of the following data evaluation external sources:

(a) Human producible outputs.

(b) Parallel corpora of source-target languages.

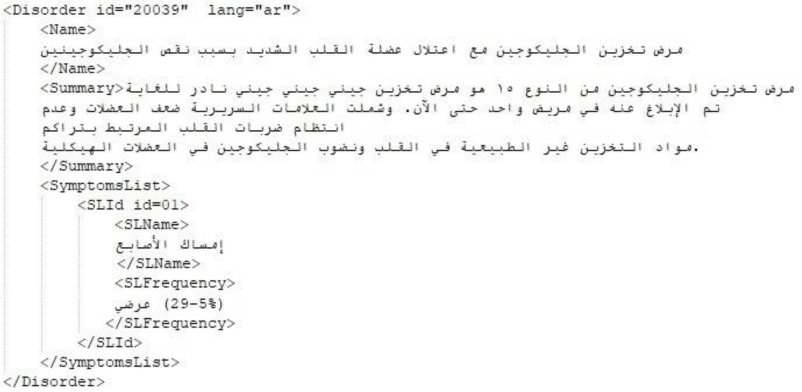
(c) Machine translation systems, such as Google Translator or Babel Fish.

Since these methods are useful for evaluating the required tasks to achieve the goal of the proposed system, we next explain them with further details. First, the qualitative method could be utilized to know the users opinion of the system's outputs. Specifically, the quality of the translation/summarization could be measured based on the users evaluation. The evaluators are provided with a set of system input and output samples. Then, this set is evaluated by assigning one of scale values. After that, the average of each chosen value is calculated. The higher value represent the quality of the task. For instance, if the higher average values is "good", the system has good quality. In this method, the users should be provided with guidelines to understand how to distinguish between the various given scale values.

Second, the BLEU Score method could be utilized to measure the accuracy of the system's outputs. In order to apply BLEU, we need to have the system's inputs, its outputs and external outputs for the system's inputs. These external outputs could be obtained from humans or other already exist systems. BLEU compares the n-grams of the system's outputs with the n-grams of the external outputs for specific inputs to calculate the number of matched grams (words). More matches means better evaluation.

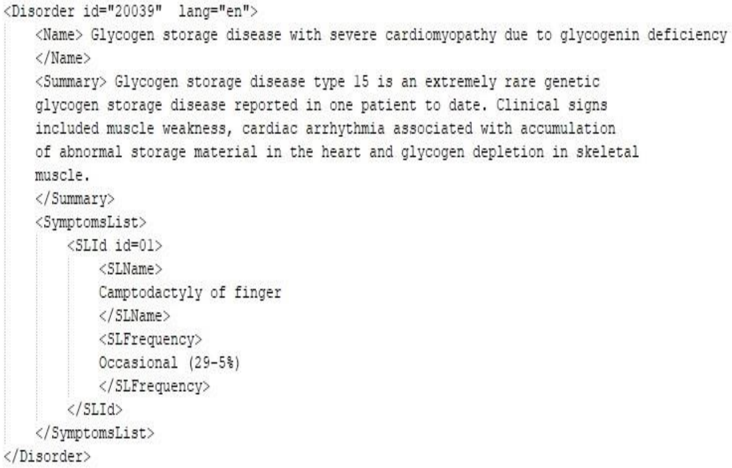
Since obtaining human outputs of summarization and translation is time consuming and costly, this research proposes evaluating the system's outputs with the outputs of already evaluated exist systems (summarization generators and machine translators). However, this limits the evaluation process because it limits the evaluation of the system's outputs to the the exist systems' outputs. This means that it could show less or equal quality outputs but not better. For instance, Figure 5 shows a possible external output using Google Translator for a randomly selected system's input from the English summaries corpus, see Figure 6.

Thus, BLEU could be applied as a first step of the evaluation; second, a qualitative method could be applied to compare the system's outputs to the external outputs via human judgments. In this method, the evaluators compare a given set that consists of the inputs, the system's outputs, and the external outputs to select the better outputs. If the system's outputs has more selections, it means their quality is better.



```
<Disorder id="20039"  lang="ar">
    <Name>
    مرض تخزين الجليكوجين مع اعتلال عضلة القلب الشديد بسبب نقص الجليكوجينين
    </Name>
    <Summary>مرض تخزين الجليكوجين من النوع ١٥ هو مرض تخزين جيني جيني جيني نادر للغاية
    تم الإبلاغ عنه في مريض واحد حتى الآن. وشملت العلامات السريرية ضعف العضلات وعدم
    انتظام ضربات القلب المرتبط بتراكم
    مواد التخزين غير الطبيعية في القلب ونضوب الجليكوجين في العضلات الهيكلية.
    </Summary>
    <SymptomsList>
        <SLId id=01>
            <SLName>
            إمساك الأصابع
            </SLName>
            <SLFrequency>
            عرضي (29-5%)
           </SLFrequency>
        </SLId>
    </SymptomsList>
</Disorder>
```

Figure 5. Arabic Summaries Corpus Sample.



```
<Disorder id="20039"  lang="en">
    <Name> Glycogen storage disease with severe cardiomyopathy due to glycogenin deficiency
    </Name>
    <Summary> Glycogen storage disease type 15 is an extremely rare genetic
    glycogen storage disease reported in one patient to date. Clinical signs
    included muscle weakness, cardiac arrhythmia associated with accumulation
    of abnormal storage material in the heart and glycogen depletion in skeletal
    muscle.
    </Summary>
    <SymptomsList>
        <SLId id=01>
            <SLName>
            Camptodactyly of finger
            </SLName>
            <SLFrequency>
            Occasional (29-5%)
            </SLFrequency>
        </SLId>
    </SymptomsList>
</Disorder>
```

Figure 6. English Summaries Corpus Sample.

5. Conclusion

This paper presents the planed work for English-Arabic medical texts translation and summarization using neural network techniques. Future work would be implementing these techniques to collect data, report the outputs and measure the accuracy of the system.

6. Acknowledgement

Thanks for Imam Abdulrahman Bin Faisal University for financing this work as part of PhD degree in Leeds University.

7. References

1. Meilleur, K. G. & Littleton-Kearney, M. T. Interventions to improve patient education regarding multifactorial genetic conditions: a systematic review. *American Journal of Medical Genetics Part A* 149, 819–830 (2009).
2. Howerton, D. A. Medical Information on the Internet. *Journal of Pastoral Care & Counseling* 73, 52–54 (2019).

3. Bustard, D. & Liu, W. *Soft-Ware 2002: Computing in an Imperfect World: First International Conference, Soft-Ware 2002 Belfast, Northern Ireland, April 8-10, 2002 Proceedings* (Springer, 2003).
4. Silla, C. N., Pappa, G. L., Freitas, A. A. & Kaestner, C. A. *Automatic text summarization with genetic algorithm-based attribute selection* in *Ibero-American Conference on Artificial Intelligence* (2004), 305–314.
5. Jaykumar, N. *ResQu: A Framework for Automatic Evaluation of Knowledge-Driven Automatic Summarization* PhD thesis (Wright State University, 2016).
6. Mantas, J., Hasman, A. & Househ, M. S. *Enabling Health Informatics Applications* (IOS Press, 2015).
7. Afantenos, S., Karkaletsis, V. & Stamatopoulos, P. Summarization from medical documents: a survey. *Artificial intelligence in medicine* 33, 157–177 (2005).
8. Moratanch, N. & Chitrakala, S. *A survey on abstractive text summarization* in *2016 International Conference on Circuit, power and computing technologies (ICCPCT)* (2016), 1–7.
9. Mishra, R. *et al.* Text summarization in the biomedical domain: a systematic review of recent research. *Journal of biomedical informatics* 52, 457–467 (2014).
10. Moawad, I. F. & Aref, M. *Semantic graph reduction approach for abstractive Text Summarization* in *2012 Seventh International Conference on Computer Engineering & Systems (ICCES)* (2012), 132–138.
11. Le, H. T. & Le, T. M. *An approach to abstractive text summarization* in *2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR)* (2013), 371–376.
12. Zhang, H., Fiszman, M., Shin, D., Wilkowski, B. & Rindflesch, T. C. Clustering cliques for graph-based summarization of the biomedical research literature. *BMC bioinformatics* 14, 182 (2013).
13. Bhargava, R., Sharma, Y. & Sharma, G. Atssi: Abstractive text summarization using sentiment infusion. *Procedia Computer Science* 89, 404–411 (2016).
14. Khan, A. *et al.* Abstractive text summarization based on improved semantic graph approach. *International Journal of Parallel Programming* 46, 992–1016 (2018).
15. Kishore, K., Gopal, G. N. & Neethu, P. *Document Summarization in Malayalam with sentence framing* in *2016 International Conference on Information Science (ICIS)* (2016), 194–200.
16. Azadani, M. N., Ghadiri, N. & Davoodijam, E. Graph-based biomedical text summarization: An itemset mining and sentence clustering approach. *Journal of biomedical informatics* 84, 42–58 (2018).
17. Gigioli, P., Sagar, N., Rao, A. & Voyles, J. *Domain-Aware Abstractive Text Summarization for Medical Documents* in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2018), 2338–2343.
18. Yao, K. *et al.* Dual encoding for abstractive text summarization. *IEEE transactions on cybernetics* (2018).
19. Jose, J. M. *et al. Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II* (Springer Nature, 2020).
20. Song, S., Huang, H. & Ruan, T. Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications* 78, 857–875 (2019).

21. Iwasaki, Y., Yamashita, A., Konno, Y. & Matsubayashi, K. *Japanese abstractive text summarization using BERT* in *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)* (2019), 1–5.
22. Sotudeh, S., Goharian, N. & Filice, R. W. Attend to Medical Ontologies: Content Selection for Clinical Abstractive Summarization. *arXiv preprint arXiv:2005.00163* (2020).
23. Hassan, S. & Mihalcea, R. *Cross-lingual semantic relatedness using encyclopedic knowledge* in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (2009), 1192–1201.
24. Lee, Y.-Y., Ke, H., Huang, H.-H. & Chen, H.-H. *Combining word embedding and lexical database for semantic relatedness measurement* in *Proceedings of the 25th International Conference Companion on World Wide Web* (2016), 73–74.
25. Navigli, R. & Ponzetto, S. P. *BabelRelate! a joint multilingual approach to computing semantic relatedness* in *Twenty-Sixth AAAI Conference on Artificial Intelligence* (2012).
26. Bhingardive, S., Redkar, H., Sappadla, P., Singh, D. & Bhattacharyya, P. *Indowordnet:: similarity computing semantic similarity and relatedness using indowordnet* in *Global WordNet Conference* (2016), 39.
27. Camacho-Collados, J., Pilehvar, M. T., Collier, N. & Navigli, R. *Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity* in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (2017), 15–26.
28. Speer, R. & Lowry-Duda, J. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. *arXiv preprint arXiv:1704.03560* (2017).
29. Yu, Z., Wallace, B. C., Johnson, T. & Cohen, T. Retrofitting concept vector representations of medical concepts to improve estimates of semantic similarity and relatedness. *Studies in health technology and informatics* 245, 657 (2017).
30. Abdedda"ım, S., Vimard, S. & Soualmia, L. F. The MeSH-gram Neural Network Model: Extending word embedding vectors with MeSH concepts for UMLS semantic similarity and relatedness in the biomedical domain. *arXiv preprint arXiv:1812.02309* (2018).
31. Henry, S., Cuffy, C. & McInnes, B. T. Vector representations of multi-word terms for semantic relatedness. *Journal of biomedical informatics* 77, 111–119 (2018).
32. Heo, G. E. & Xie, Q. *A Hybrid Semantic Relatedness Algorithm by Entity CoOccurrence and Specialized Word Embeddings* in *2019 IEEE International Conference on Healthcare Informatics (ICHI)* (2019), 1–2.
33. Glasgow, K., Roos, M., Haufler, A., Chevillet, M. & Wolmetz, M. Evaluating semantic models with word-sentence relatedness. *arXiv preprint arXiv:1603.07253* (2016).
34. Siblini, R. & Kosseim, L. CLaC: Semantic relatedness of words and phrases. *arXiv preprint arXiv:1708.05801* (2017).
35. He, H., Gimpel, K. & Lin, J. *Multi-perspective sentence similarity modeling with convolutional neural networks* in *Proceedings of the 2015 conference on empirical methods in natural language processing* (2015), 1576–1586.

36. Tian, J., Zhou, Z., Lan, M. & Wu, Y. *Ecnu at semeval-2017 task 1: Leverage kernelbased traditional nlp features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity* in *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)* (2017), 191–197.

37. GOMAA, W. H. A MULTI-LAYER SYSTEM FOR SEMANTIC RELATEDNESS EVALUATION. *Journal of Theoretical and Applied Information Technology* 97 (2019).

38. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I. & Specia, L. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv: 1708.00055* (2017).

39. Hu, Y., Ye, X. & Shaw, S.-L. Extracting and analyzing semantic relatedness between cities using news articles. *International Journal of Geographical Information Science* 31, 2427–2451 (2017).

40. Khan, M., Ramzan, S., Khan, S., Hassan, S. & Saeed, K. Measuring Text-Based Semantics Relatedness Using WordNet. *International Journal of Cognitive and Language Sciences* 13, 316–319 (2019).

41. Al-Ajmi, H. A new English–Arabic parallel text corpus for lexicographic applications. *Lexikos* 14 (2004).

42. Alotaibi, H. M. Arabic-English parallel corpus: a new resource for translation training and language teaching. *Arab World English Journal (AWEJ) Volume* 8 (2017).

43. Zeroual, I. & Lakhouaja, A. MulTed: A multilingual aligned and tagged parallel corpus. *Applied Computing and Informatics* (2020).

44. Ahmad, A. A.-S., Hammo, B. & Yagi, S. ENGLISH-ARABIC POLITICAL PARALLEL CORPUS: CONSTRUCTION, ANALYSIS AND A CASE STUDY IN TRANSLATION STRATEGIES. *Jordanian Journal of Computers and Information Technology (JJCIT)* 3 (2017).

45. Park, J., Kim, K., Hwang, W. & Lee, D. Concept embedding to measure semantic relatedness for biomedical information ontologies. *Journal of biomedical informatics* 94, 103182 (2019).

46. Nakamura, T., Shirakawa, M., Hara, T. & Nishio, S. Wikipedia-Based Relatedness Measurements for Multilingual Short Text Clustering. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 18, 1–25 (2018).

47. Strube, M. & Ponzetto, S. P. *WikiRelate! Computing semantic relatedness using Wikipedia* in *AAAI* 6 (2006), 1419–1424.

48. Morgan, J. T. *et al.* Are we there yet?: The development of a corpus annotated for social acts in multilingual online discourse. *Dialogue & Discourse* 4, 1–33 (2013).

49. Kim Jung, J. H. *Gender bias in natural language processing: BioCorpus-5, a preliminary multilingual Gender-Balanced Corpus of in-domain wikipedia biographies* B.S. thesis (Universitat Politécnica de Catalunya, 2019).

50. Frej, J., Schwab, D. & Chevallet, J.-P. MLWIKIR: A Python toolkit for building largescale Wikipedia-based Information Retrieval Datasets in Chinese, English, French, Italian, Japanese, Spanish and more.

51. Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606* (2016).

52. Řehůřek, R. & Sojka, P. *Software Framework for Topic Modelling with Large Corpora* English. in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* http://is.muni.cz/publication/884893/en (ELRA, Valletta, Malta, May 2010), 45–50.
53. Navigli, R. & Ponzetto, S. P. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence* 193, 217–250 (2012).
54. Speer, R., Chin, J. & Havasi, C. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, 4444–4451. http://aaai.org/ocs/index.php/AAAI/AAAI17/ paper/view/14972 (2017).
55. Goldberg, Y. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies* 10, 1–309 (2017).
56. Farzad, A., Mashayekhi, H. & Hassanpour, H. A comparative performance analysis of different activation functions in LSTM networks for classification. *Neural Computing and Applications* 31, 2507–2521 (2019).
57. Cho, K. *et al.* Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
58. Cohan, A. *et al.* A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685* (2018).
59. Yang, S., Wang, Y. & Chu, X. A Survey of Deep Learning Techniques for Neural Machine Translation. *ArXiv* abs/2002.07526 (2020).
60. Schuster, M. & Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 2673–2681 (1997).
61. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* 9, 1735–1780 (1997).
62. Gers, F. A., Schmidhuber, J. & Cummins, F. Learning to forget: Continual prediction with LSTM (1999).
63. Xu, K. *et al. Show, attend and tell: Neural image caption generation with visual attention* in *International conference on machine learning* (2015), 2048–2057.
64. Kalchbrenner, N. & Blunsom, P. *Recurrent continuous translation models* in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (2013), 1700–1709.
65. Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
66. Freitag, M. & Al-Onaizan, Y. Beam search strategies for neural machine translation. *arXiv preprint arXiv:1702.01806* (2017).
67. Nwankpa, C., Ijomah, W., Gachagan, A. & Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378* (2018).
68. Nguyen, G. *et al.* Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review* 52, 77–124 (2019).
69. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12, 2825–2830 (2011).
70. Oliphant, T. E. Python for scientific computing. *Computing in Science & Engineering* 9, 10–20 (2007).
71. Bird, S., Klein, E. & Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit* (" O'Reilly Media, Inc.", 2009).

72. Řehůřek, R. & Sojka, P. Gensim—statistical semantics in python. *Retrieved from genism. org* (2011).

73. Srinivasa-Desikan, B. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras* (Packt Publishing Ltd, 2018).

74. McKinney, W. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython* (" O'Reilly Media, Inc.", 2012).

75. Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J. *BLEU: a method for automatic evaluation of machine translation* in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (2002), 311–318.

76. Dew, K. N., Turner, A. M., Choi, Y. K., Bosold, A. & Kirchhoff, K. Development of machine translation technology for assisting health communication: A systematic review. *Journal of biomedical informatics* 85, 56–67 (2018).

Copyright Disclaimer