Expanding Horizons
Business, Management and Technology for Better Society
20–22 May 2020
Online Conference

Management, Knowledge and Learning International Conference 2020
Technology, Innovation and Industrial Management

Assessing American Presidential Candidates Using Principles of Ontological Engineering, Word Sense Disambiguation, and Data Envelope Analysis

James Rodger
IUP, United States
jrodger@iup.edu

Abdullah Murrar
IUP, United States
khkbc@iup.edu

Pankaj Chaudhary
IUP, United States
pankaj@iup.edu

Brandon Foley
IUP, United States
xghz@iup.edu

Marouane Balmakhtar
IUP, United States
sxjz@iup.edu

Justin Piper
Penn State New Kensington, United States
jmp7152@psu.edu

Abstract

Word Sense Disambiguation (WSD) is the process of automatically identifying which the appropriate meaning of a word given in its sentence. (WSD) is a promising research area in computational linguistics, especially in wide range of advanced applications, such as medical and social sciences. This research employs the concept (WSD) to determine the inherent meaning of voter intentions regarding possible political candidates. Where candidates can be examined and their true assets and competencies in three major areas of eligibility, education, *and experience inputs can be deciphered. Data envelope analysis (DEA) is used to determine underlying word instances for elected and successful outputs. The results demonstrate the validity of using (DEA) as a tool for (WSD). The results also indicate that the survey administered by the website which is developed for the purpose of this research, and used in this study, is a promising tool for predicting successful presidential candidates.*

Keywords: Artificial Intelligence, Data Envelope Analysis, Ontological Engineering, Word-Sense Disambiguation

Introduction

A word can be interpreted in several ways depending on the context in which it exists. Word Sense Disambiguation (WSD) is the task of automatically identifying the appropriate meaning of a word given in its context. WSD has received increasing attention in many fields such as applications of marketing and customer behavior (Placeholder1), biomedical domain (Duque et al, 2018), machine translation (Jianping Yu et al, 2016), email spam filtering (Laorden et al, 2012), information retrieval (Chifu et al, 2015) and other applications. Word Sense Disambiguation can be used also in improving government systems and performance, for instance, applying the concept to help citizen-voters strengthen their ability to analyze and evaluate the candidate and incumbent performance with a standard approach and criteria. The candidate can be for positions of village council, mayor, governor or country president.

This paper investigates the expression “good American president” as an instance of word-sense disambiguation (WSD). Making a prediction of who could become a “good American president” is founded on information discovered while vetting a candidate. Advanced technology and tools are used to support voters in a democratic republic, as their responsibility of choosing their government. A good citizen must be actively involved in government at every level, beginning in the town, city, or county, continuing through the district and state, and ultimately finishing with the federal government. It would be a disservice to democracy to make good citizenship sound easy. The denial of adequate support by political parties to voters undermines free rule to an equal extent. That is why research of this kind is essential to improve the system.

This research project mixes in its design between the practice and theory. The researchers evaluate three acting entities: candidate, voter, and website. The project also is conducted to assess and ameliorate the voters’ lack of knowledge of their preferred candidate in an upcoming election and the candidates’ lack of awareness of how well they fit voter’s needs (George & Rodger, 2017). The guiding goal of the project is to determine how to select the next American president. The first step was to create a set of questionnaires for both the candidate and the voter. These questionnaires are posted on a website created using ASP.NET and C#, and they are to be completed by the candidates first, followed by the voters.

The questions for the candidates are divided into three sections. The first is Legal Eligibility. In this eliminator section, the determination is whether the candidate is eligible to contest the election. If not, the candidate is eliminated and unable to respond the next section of questions. The second section is Basic Bio-data. Here, the candidates’ resumes are built through questions on their background, from their name and birth place, to educational qualification and alma mater, to occupational and work history, among other items. This section is useful for voters to learn about the backgrounds of the candidates.

The third section is an Intellectual Assessment. This round of questions establishes the knowledge and policies of the candidates in different domain areas (including the environment, education, terrorism, health care, research and development, gun violence, inflation, corruption, and racism) that they would need to face as president. The third section is important because a voter might only be considering candidates for president in terms of a limited set of responses to a given range of concerns, while the candidates might be thinking in different terms or might not address the voter's pet issues at all.

The responses to this set of questions would benefit both the voter and candidate. In this way, each voter would have a chance to understand the unique thinking of each candidate that will guide that candidate if given a chance to lead the country as president. For their part, in this way, the candidates could come to understand what the voters expect from them, using the equivalent portion of the voter survey. Once the candidates finish their responses, the voters will receive their questionnaires.

Of course, the voters' questionnaire is not as complex as the candidates' and is more generic but covers the same domain areas as the candidates' questionnaire. The voters here have the same privilege as the candidates of prioritizing the questions. Once the voters have completed their questionnaires, the website will process the answers and begin matching voters' responses to those provided by the candidates.

This project examines this in specific, concrete terms. The challenge is daunting, even in simplified examples. However, the harnessed power of information technology, data, mathematical analysis, and algorithms can bring understanding within reach. The researchers implemented a Data Envelop Analysis (DEA) to provide weighted averages to match the priorities assigned for different questions by the voters and the candidates to indicate which candidate best matches a given voter's answers.

This method is used to match the asymmetrical data provided by the two groups. For every question the voter answers, there is more than one question on the same subject answered by the candidate. This is done to obtain more detailed information on the candidate's perspective on the given topic. Because the voters answer fewer questions than the candidates, the matching algorithm must be complex. The weighted average method helps compute the average for all candidates' answers and matches the averages for every question answered by the voter. This will act as a self-assessment for the candidate and a voter–candidate compatibility calculator for the voter.

Materials and Methods

There are several approaches discussed and fields tacked in the literature about Word Sense Disambiguation (WSD). Amy Neustein (2012) analyzed three different areas of natural language processing (NLP) from a cognitive science perspective, namely, simulation of human language use in spoken dialogue systems, reference generation and referential practices, and WSD. Studies in subjects as varied as finance and medicine have used extractive summation, sentiment classification, and corpus-based stemming techniques, which are taken from this field of study (Arroyo-Fernández et al, 2019) (Singh & Gupta, 2019), (Jie Xu et al, 2019), (Frank Xing et al, 2019).

Oi YeeKwong (2008) examined the psychological evidence for internal lexicons and suggested that concrete senses are more readily activated than abstract ones and that broad associations are more easily triggered than narrow paradigmatic ones. Bateman (2010) reviewed recently developed approaches to the semantics of natural language expressions that drew on a new combination of the principles of ontological engineering and natural dialogue system behavior involving spatial information. Silke Maren Witt (2015) proposed a probabilistic response time model to calculate the likelihood of user response at any time that can be used for timeout setting optimization.

Lakhfif & Laskri (2016) described the sign and implementation of a computational model for an Arabic semantic parser that could create a deeper semantic representation of Arabic text. They showed that the integration of WordNet and FrameNet can improve disambiguation accuracy. AlMaayah et al (2016) developed automatic extraction model for synonyms, which they used to construct their Quranic Arabic WordNet, which uses traditional Arabic dictionaries. This project improved the recall of semantic search for Quranic concepts by 27%.

Chengyao Lv et al (2016) described speech (POS) tagger, used to classify unannotated natural language words with POS labels for categories such as noun, verb, and adjective. Kadim & Lazrek (2016) proposed a hypothesis for the selection results for a POS tagging implemented for the Arabic language and presented numerous cases where the morphosyntactic state of a word depends on the states of the subsequent words. Marwah Alian et al (2016) introduced a to Arabic WSD, utilizing Wikipedia as a lexical resource for disambiguation. This approach was also tested on English words for improved generalizability. El Mahdaouy et al (2018) developed a method to incorporate word embedding semantic similarities into existing probabilistic information retrieval (IR) models for Arabic to deal with term mismatch. Their results showed that extending the IR model can improve the baseline bag-of-words model and that their extensions significantly improve the Arabic WordNet-based semantic indexing approach.

(Menai (2014) found that the genetic algorithms can achieve more precise prediction than the memetic algorithms and the naïve Bayes classifier by carrying out on a large Arabic corpus. While (Abualhaija & Zimmermann, 2016) found that bee and ant colony optimization perform better results than the genetic and simulated annealing algorithms. Boudchiche & Mazroui (2019) created an Arabic lemmatization, including two modules. They adopted hidden Markov models and validated this approach using a labeled corpus consisting of about 500,000 words.

Duque et al (2016) explored whether multilingualism can help solve problems of ambiguity or the conditions required for a system to improve the results obtained using a monolingual approach in WSD. They determined the optimal means of generating those useful multilingual resources, and they studied different languages and sources of knowledge. Andres Duque et al (2018) presented a new graph-based, unsupervised technique to address this problem. They used a knowledge base in the form of a graph built from co-occurrence information on medical concepts as found in scientific abstracts. Izquierdo & Su ´ arez (2009) used a very simple method of deriving a small set of appropriate meanings employing the basic structural properties of WordNet. They empirically demonstrated that this automatically derived set of meanings groups senses into an adequate level of abstraction to perform class-based WSD, with 80% accuracy. Singha & Siddiquia (2015) investigated the role of hypernym, hyponym, holonym, and meronym relationships in Hindi using WSD. They found that maximum improvement in single semantic relationships was obtained using hyponyms, which resulted in an overall improvement in precision of 9.86%.

Hung & Chen (2016) examined three techniques for WOM documents to build WSD-based SentiWordNet lexicons, and their experiments demonstrated that the results improved with this lexicon. Lopez-Arevalo et al (2017) described an approach to domain-specific WSD by selecting a predominant sense for ambiguous words, using two corpora, a domain-specific test corpus and a domain-specific auxiliary corpus, and they tested their approach on sports and finance texts. CorrêaJr et al (2018) focused on semantic relationships and represented texts as graphs, but they also constructed a structure through which sense discrimination can be achieved. Their learning algorithm outperformed the support vector machine algorithm in particular cases.

So far, Word Sense Disambiguation has long been considered as one of the most complex research fields, and it is at the same time a promising research areas in computational linguistics. The current progress in its resolution would envision great advances in NLP-based applications. As developed from content-based e-mail filtering, document search and retrieval, to subject-based document classification, semantic text summarization, and question answering in wide range of advanced applications in medical, behavioral, and social. In all the application fields, there is a need for software packages that collect, process and store the data. One of the components of this research project is developing a website that practically allows candidates and voters to answer the survey questions, saves the answers in the database, and displays the results. The design of the software and the ontological architecture can be detailed in the theory and design section as in the next.

Theory and Design

The researchers followed the Unified Modeling Language (UML) principles in the website design as this language is widely used in industry and academia for the analysis and design the information systems, thus, guarantee a scientific framework for the design and analysis (Bork et al, 2019). This process is initiated by summarizing a user's interaction with a system that demonstrates the relationship between the user and the particular use case in effect. The use case diagram of this presidential candidate project contains the actors, administrator, database, voter, and candidate. Each actor is assigned its own associations and events that pertain to it individually.

The administrator feeds candidate data, prepares simple questionnaires for voters and complex questionnaires for candidates, and modifies the questionnaires. The voters enter the website, answer the simple questionnaires, and obtain the results. The candidates also enter the websites, answer the complex questionnaires, and obtain the results. The database user stores the candidate's data, stores the login information, stores the questionnaires as well as stores the results.

Events and Activities

An event is something that takes place at a particular time and place that can be portrayed and merits recalling (Bhattacharyya et al, 2018). This concept is used in system analysis and design. Any complex framework should be broken into reasonable units to make it comprehensible, and the system should be decomposed, based on events. An event table lists events in tabular format, with each event at the head of a row and key pieces of information for each in columns. The event table for the project is given below.

Table 1: Events Table

Event	Trigger	Activity	Destination
User opens website	Load webpage and input candidate qualities	Determines if inputs allow the user to start the questionnaire	User
User completes Part A of questionnaire	User inputs answers for Parts B and C of questionnaire	Answers are collected to be sent to historical data to be compared and analyzed	User
Questionnaire	Questionnaire	Answers are collected to be sent to historical data to be compared and analyzed	User
Analysis of user and historical data	User clicks 'submit' to send data to be compared	All input is sent to historical data to determine candidacy by comparisons and analysis	Historical data
Send user results	User clicks 'see results'	Returns analysis results and potential for candidacy in a form	User

State Diagram

A state chart diagram is a type of diagram used in computer science to describe the behavior of systems. The state chart is one of the five types of UML diagrams used to model the dynamic nature of a system (ItI Education, 2010). These define the various states of an object during its lifetime, where the states are changed by events. Thus, state chart diagrams are useful for modeling reactive systems, which can be defined as those that respond to external or internal events, such as, in this case, responses to our website.

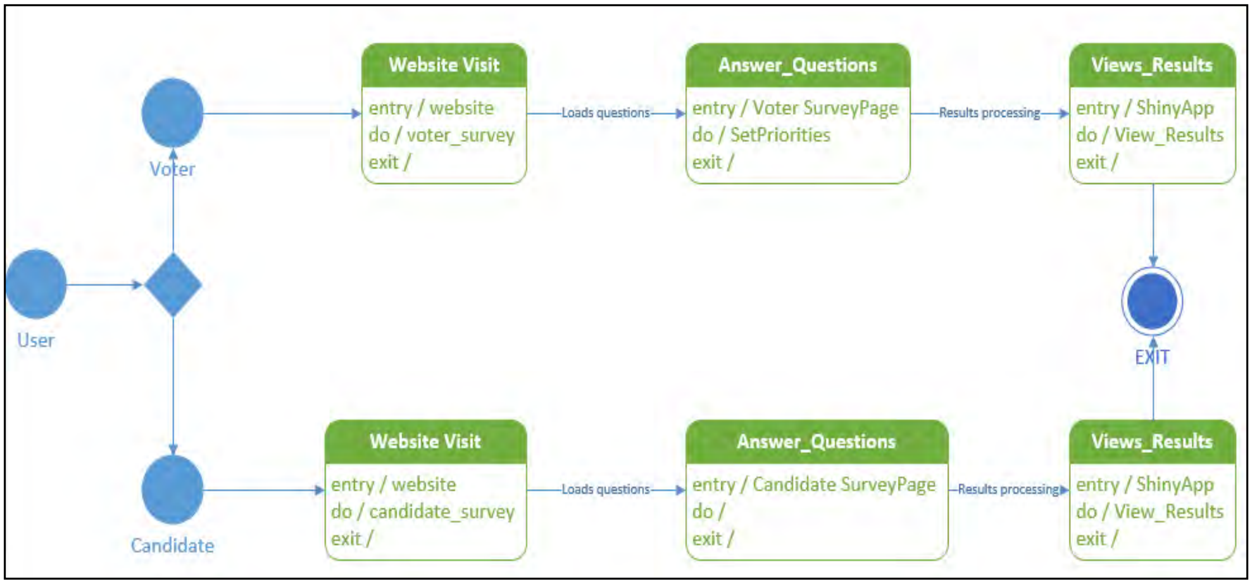

Fig. 1. State Chart Diagram

The state chart diagram in Figure 1 classifies users into two types: voter and candidate, and shows the flow of system processes. Every action of the user is defined as a function and is called in the state chart diagram. A Create, Read, Update, and Delete (CRUD) diagram is seen in Figures 2 and 3. Those figures demonstrate the connections among procedures and information or a mong procedures and assets. Where a connection exists indicates whether the procedure performs one of the CRUD operations on the information or asset. CRUD is relevant for the user interface of most applications.

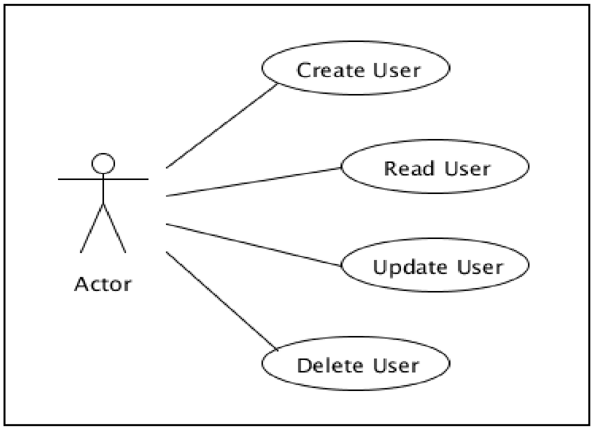

Fig. 2. CRUD use case 1

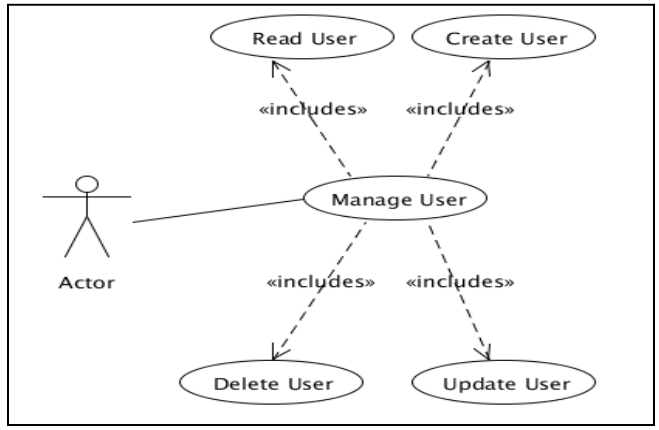

Fig. 3. CRUD use case 2

In the design of this project, CRUD was used to group data entities into subject databases, to group business functions or processes into business areas, and to identify areas that should be examined for problems, such as data entities not created or used by any business function or process, business functions or processes that do not access any data entities, and data entities created by more than one business function or process.

Table 2: CRUD Table for the How to Select an American President Website

Number	User action	Create	Read	Update	Delete
1	User opens website				
2	User completes Part A of questionnaire	X	X	X	
3	User completes Parts A and B of questionnaire	X	X	X	
4	Analysis of user and historical data		X		
5	Send user results		X	X	

Activity Diagram

An activity diagram is portrayed in Figure 4. It is another important diagram for UML, used to describe the dynamic aspects of the system by forming the flow of control from activity to activity among system operations in a way that can be sequential, branched, or concurrent (ItI Education, 2010). For our project, the candidates and voters enter the website, are redirected to the survey page, respond to the questions, and submit their answers. These answers are then matched using the algorithm to produce relevant results. These results are graphically displayed with the R Package Shiny.

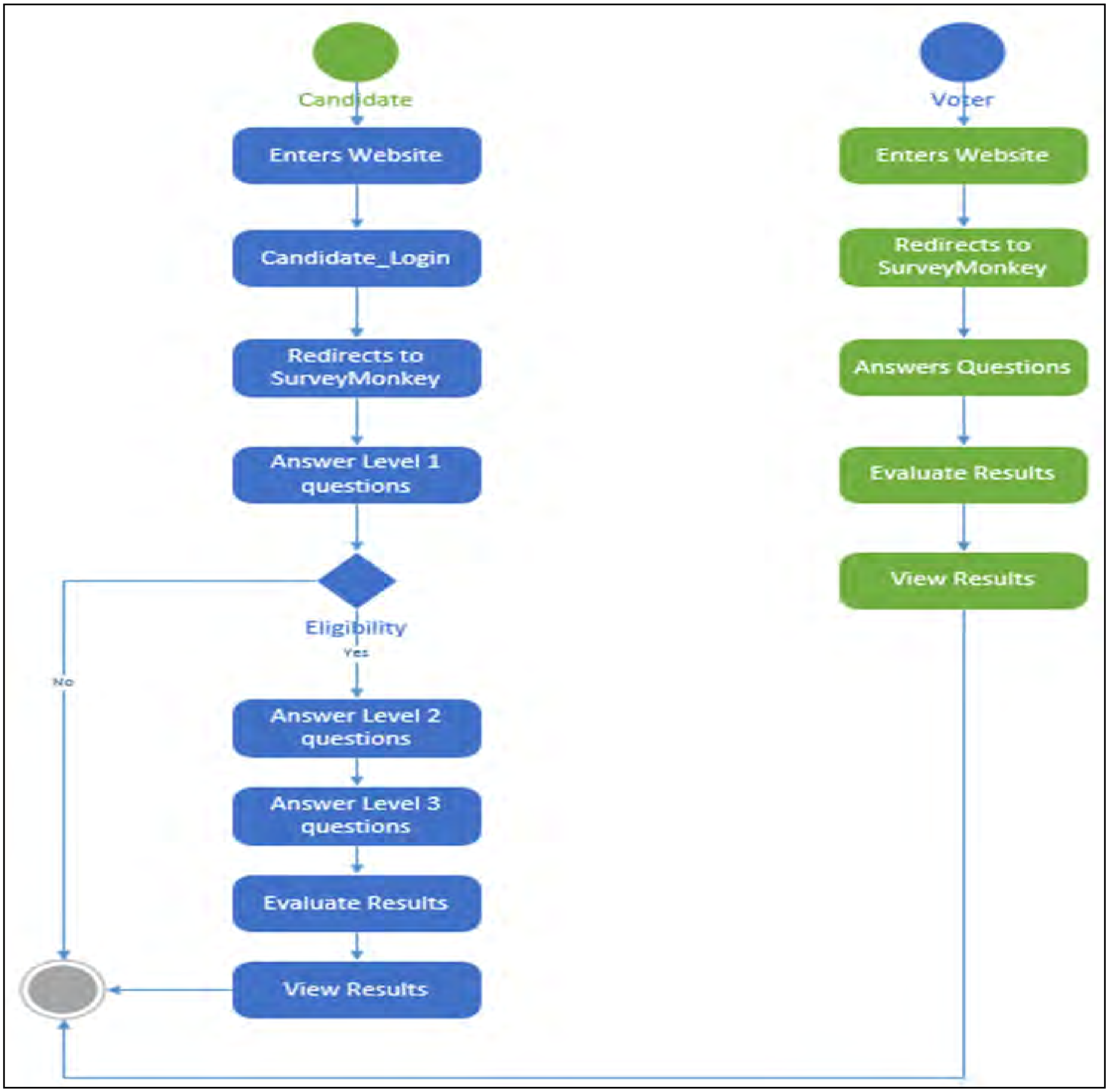

Fig. 4. Activity Diagram

Sequence Diagram

A sequence diagram interactively represents how processes work with one another and in what order, a message sequence chart construct (Donald Bell, 2004). In Figure 5, the sequence diagram shows object interactions arranged in a time sequence. It depicts the objects and classes involved in a scenario and the sequence of messages that must be exchanged between the objects to develop the functionality of the scenario. For this project, the objects on the lifelines are the candidate, voter, server, website, and database.

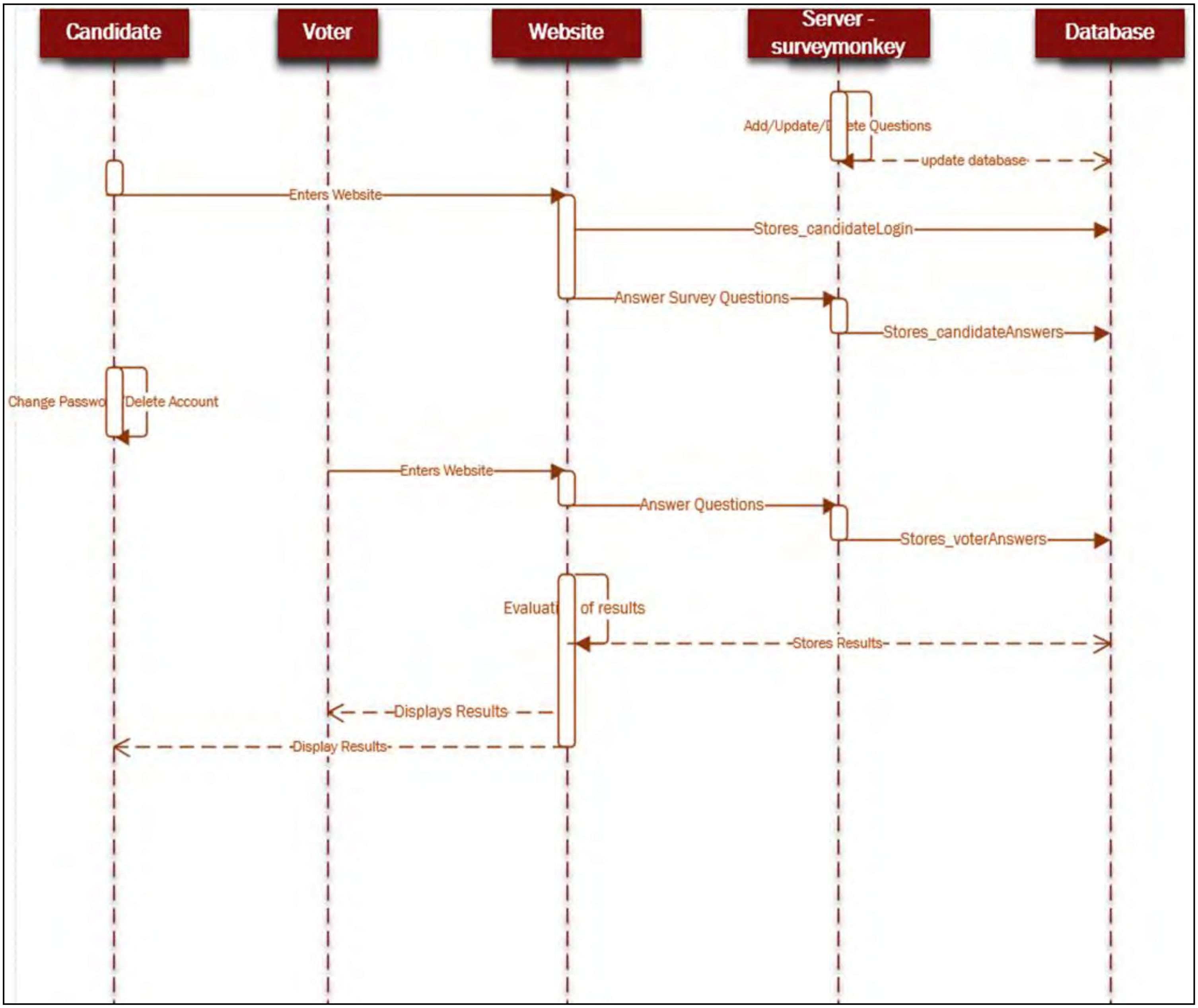

Fig. 5. Sequence Diagram

Entity Relationship Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database (ItI Education, 2010). An entity in our context is a datum, so here, an ERD illustrates the logical structure of a database. An ERD can be used to visualize how the information that a system produces is related. In Figure 6, each entity is represented by a rectangular box. The different entities, the attributes associated with each, and their different functions are clearly visible. The connections between the different entities are also clear.

Here, administrator, candidate, survey, and other items are the entities, and candidate ID, admin ID, and other elements are the attributes that describe the entities.

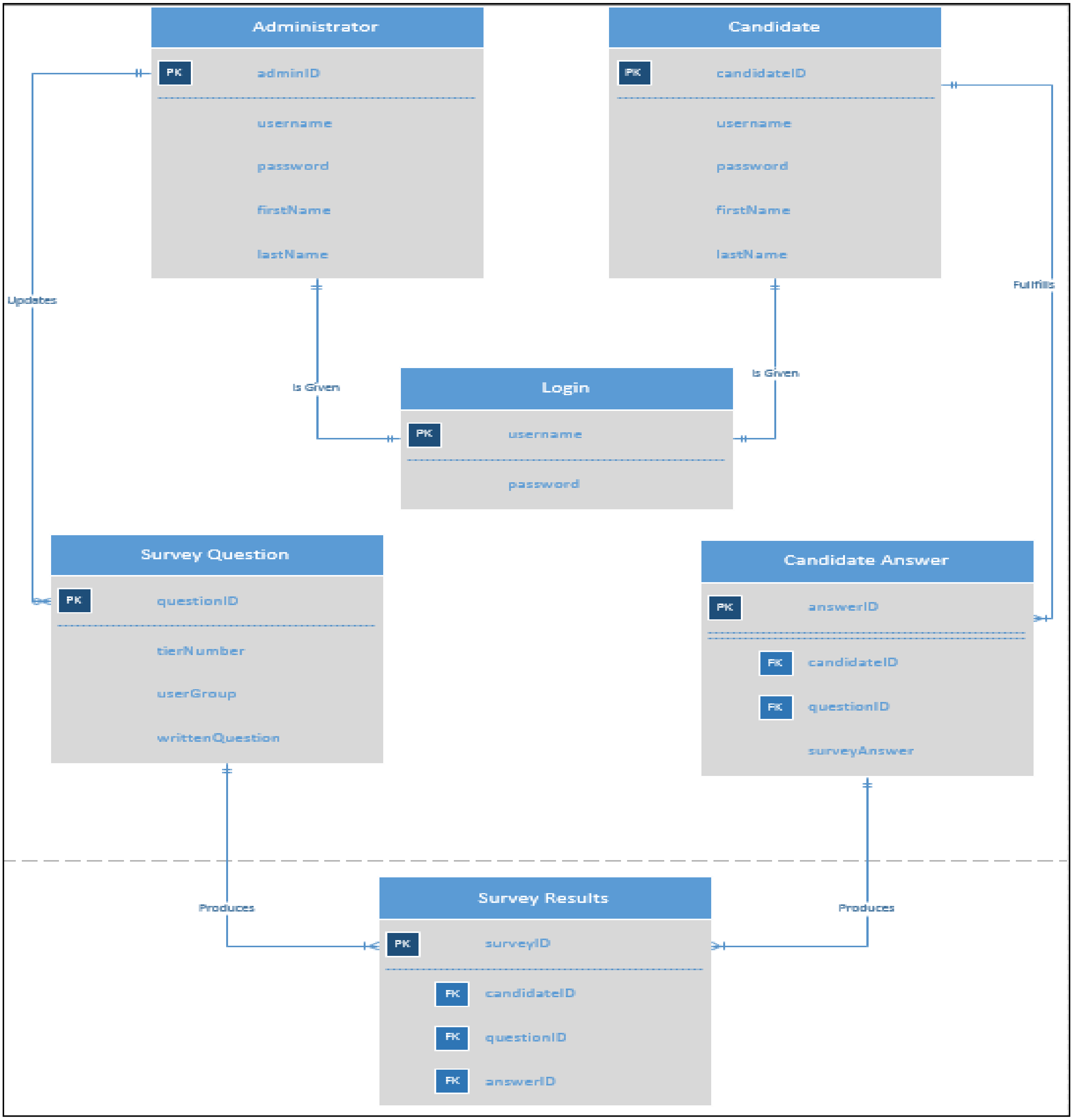

Fig. 6. Entity Relationship Diagram

Data-Flow Diagram

A data-flow diagram (DFD) is a graphic representation of how data move through an information framework (Scheel et al, 2015). In a DFD, data are shown to flow from an exterior or interior store to internal data store or an external data sink, following an inner process. DFDs include four essential segments that show how data flow within a framework: entity, process, data store, and data flow. Here, an entity is the source or destination of data.

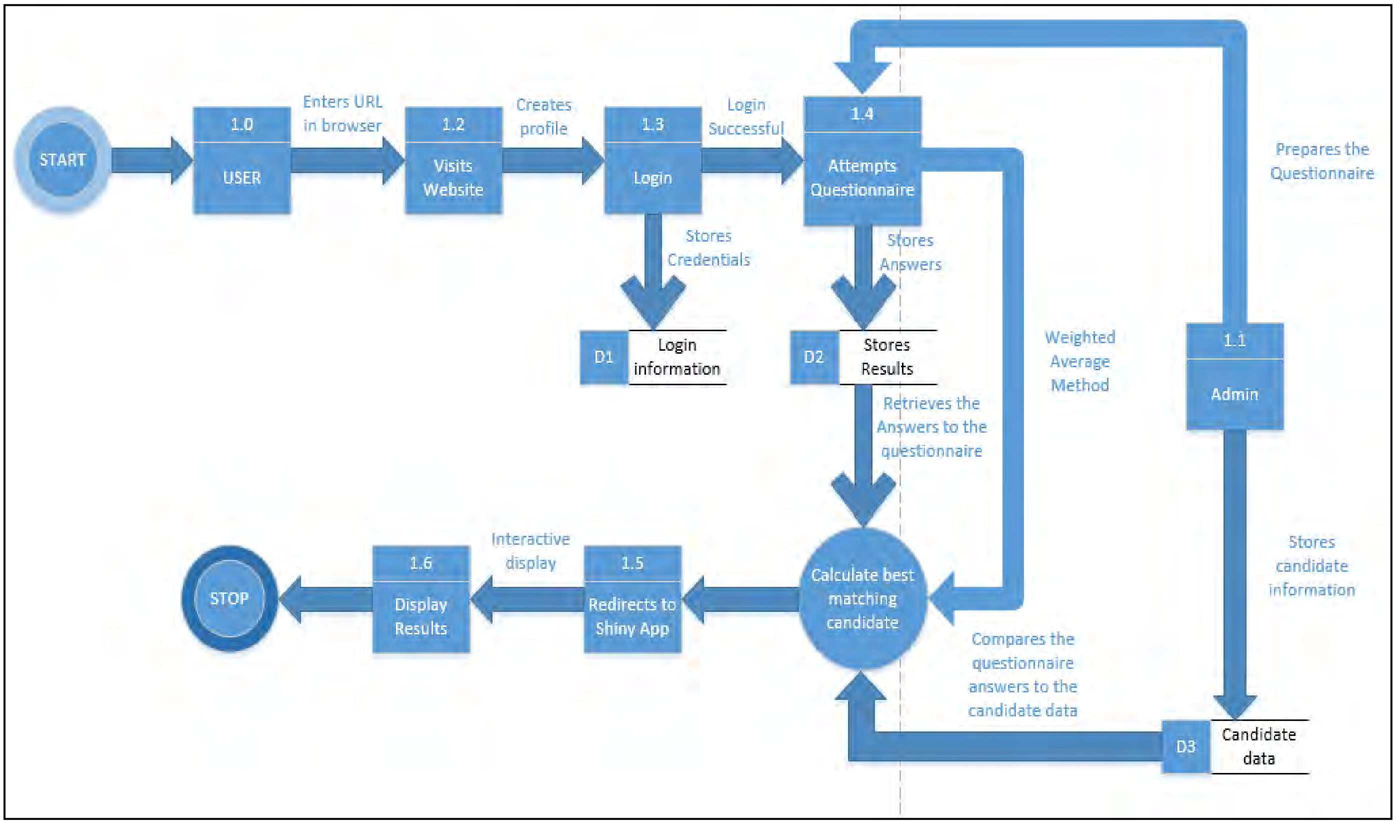

Fig. 7. Level 1 Data-Flow Diagram

Our system is broken down into many process and functions. We can see the user, login information database, results storage database, admin, candidate data stored in database, best-matching candidate process, and results display (using Shiny). Each part of this DFD is given a number from 1.0 to 1.6. The databases are named D1, D2, and D3. The admin maintains the website, adds or modifies questions, and stores candidate information in the database.

Level 2 shows a decomposition of the process of the Level 1 DFD, and as such there should be an aspect of the Level 2 DFD for every process shown in the Level 1 DFD. The Level 2 DFD is represented in Figure 8, where it is clear that it is a more refined form of the Level 1 DFD. Here, we describe the questionnaires in three sections rather than as a single block. The login block in the Level 1 DFD is broken down in the Level 2 DFD into its parts, the user ID and password sections. The remainder of the diagram remains the same. The idea of the DFD levels is to refine the diagram in subsequent levels to develop a more detailed system.

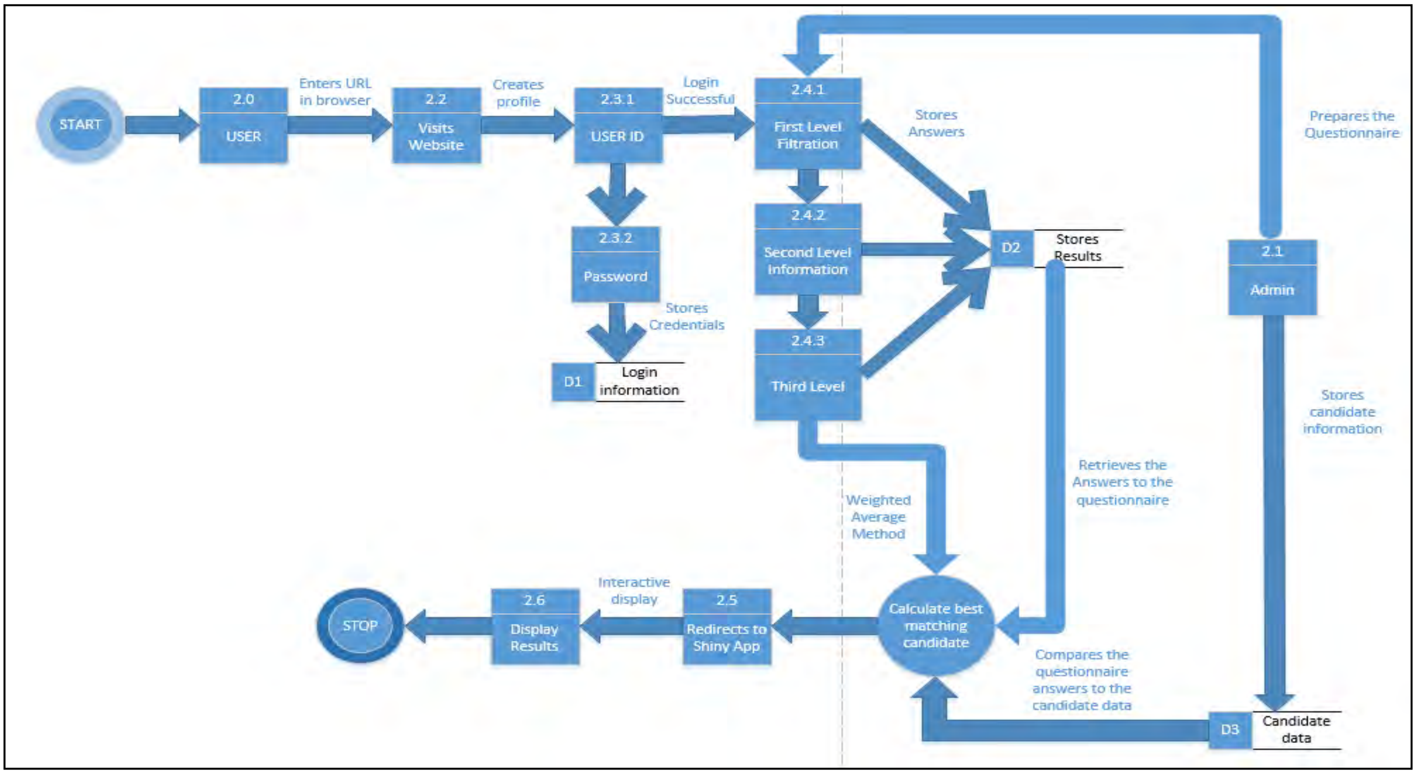

Fig. 8. Level 2 Data-Flow Diagram

Ontology Depiction

Recent studies revealed that Ontologies play key role in integrating of Software Engineering (SE) and Semantic Web Technology (SW). An ontology is a scheme of representation that describes a formal conceptualization of a domain of interest (Bhatia et al, 2016). In our case, such items are constructed using Protégé software. An ontology usually includes two distinct levels, including the intentional level, which specifies a set of conceptual elements and constraints or axioms describing the conceptual structure of the domain. The extensional level specifies the set of instances of the conceptual elements described at the intentional level. An ontology may also contain a meta-level, which specifies a set of modeling categories of which the conceptual elements are instances. The ontology of our project represents the logical flow of a process, beginning when the user logs in and ending with the user viewing the results.

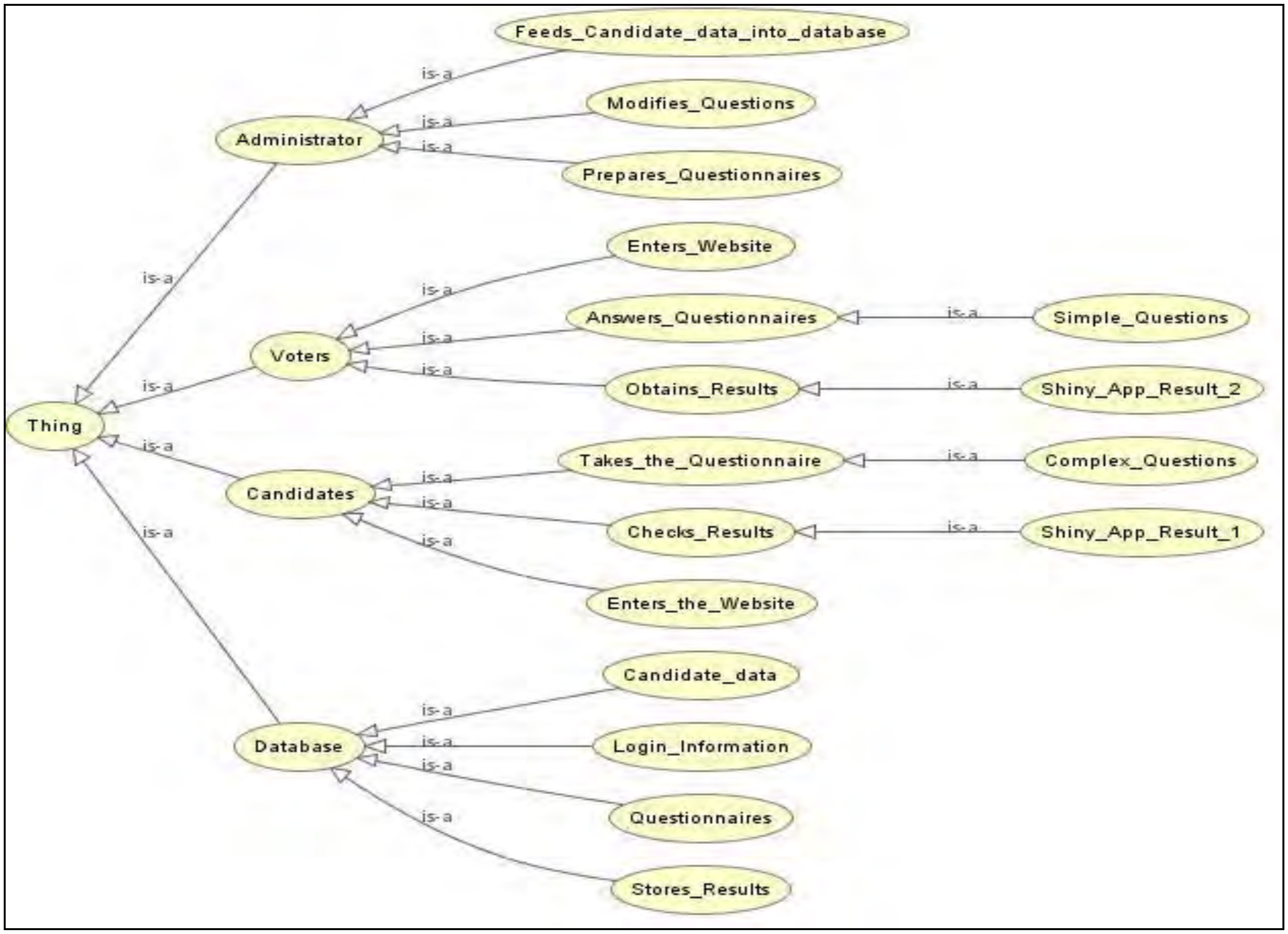

Fig. 9. Ontology Depiction Created Using Protégé

Data Dictionary

A data dictionary is a collection of descriptions of data object or item in a data model, and its relationship to other objects for reference use by programmers (ItI Education, 2010). Each data object or item is given a descriptive name, its relationships are described (or it appears as part of a structure that implicitly describes its relationships), its data type (such as text, image, or binary value) is described, possible predefined values for it are listed, and a brief textual description of it is provided. This resulting collection is then organized for reference purposes. Data dictionary of candidates table includes fields for: identity number of the candidate, first name, last name, suffix and median name.

Table 3. Candidates Table

Field name	Data type	Format	Description
id_#	Number	Auto number 3 digit	Unique key for each candidate
first_Name	Character	Varchar	Candidate's first name

middle_Name	Character	Varchar	Candidate’s middle name
last_Name	Character	Varchar	Candidate’s last name
honorary	Character	Varchar	Candidate’s honorary title or name (optional – Mr., Dr., Sir, etc.)
suffix	Character	Varchar	Candidate (candidates ending name (optional – Jr., Sr., etc.)
Maiden_Name	Character	Varchar	Candidate’s maiden or former name (optional)

There are three tables for responses. Level A response table contains field for the unique key of each candidate, and field for each question that is true and false. Level B response table is the same for level A table, but the questions are multiple choice. Level C response table includes both multiple choice, and Yes or No questions. The Response table contains all responses from the tables of levels A, B and C response tables.

Table 4. Level A Response Table

Field name	**Data type**	**Format**	**Description**
id_#	Number	Auto number 3 digit	Unique key for each candidate
q1_Response	Character	Char length 5	True or false

Table 5. Level B Response Table

Field name	**Data type**	**Format**	**Description**
id_#	Number	Auto number 3 digit	Unique key for each candidate
issue1_Response	Character	Char length 1	Choose A–D (multiple choice question regarding political standing)

Table 6. Level C Response Table

Field name	Data type	Format	Description
id_#	Number	Auto number 3 digit	Unique key for each candidate
task1.1_Response	Character	Char length 1	Choose A–E
task1.1_Comment	Character	Varchar	Optional section for candidate to add additional comments
task1.2_Response	Character	Char length 1	Choose A–D
task1.2_Comment	Character	Varchar	Optional section for candidate to add additional comments
task1.3_Response	Character	Char length 1	Choose A–C
task1.3_Comment	Character	Varchar	Optional…
task1.4_Response	Character	Char length 3	Yes/No
task1.4_Comment	Character	Varchar	Optional…

The interface of this project can be accessed through the website (www.presidentselect.com) which can be located through the survey tab of the website. Two options appear when the user hovers over the survey tab: the first one is for the candidate survey, and the second is for the voter survey.

Research Methodology

Data Envelopment Analysis (DEA) is one of the most widely used methods in the measurement of the efficiency of Decision Making Units (DMUs). There have been several studies in different fields that used the “DEA” techniques. The DEA tool was first introduced by Charnes, Cooper, and Rhodes in 1978 to develop measures of decision making efficiency for groups of entities or what called decision making units DMUs (Charnes et al, 1978). The DMUs are entities that are responsible for changing the input variable(s) into output variable(s); and therefore, the DMUs performance are to be evaluated and benchmarked (Cooper et al, 2011).

The DEA has become a very popular method in many fields. In education sector, (Gosálbez et al, 2019) assessed the efficiency of UK universities, (Aparicioa et al, 2019) scored the US school. In software packages, (Emrouznejad & Shale, 2008) combined a neural network with DEA to introduce an alternative approach to estimating the efficiency of DMUs in large datasets. Chatzigeorgiou & Stiakakis (2010), used DEA as a means of measuring and benchmarking the quality of different object-oriented software designs.

The DEA technique also implemented in other fields such as pollution emission (Zhou et al, 2019), dairy farms (Siafakas et al, 2019), hospitals (Akono et al, 2013), banking system, (Henriques et al, 2018), hotels (Sestayo & Castro, 2018), water sector (Murrar et al, 2017), (Cetrulo et al, 2019), energy performance (Wang et al, 2013), and other areas.

According to the DEA concept, a dual model can be employed to shadow prices the primary model. The constrains that limit the efficiency must be less than or equal to 1. Binding constraints have positive shadow prices, which are nonbinding at zero. If a binding constraint unit has an efficiency of 1 in the primary model, then the dual model also has a positive shadow price. The dual model is determined using the ratio of the weighted sum of the outputs to the weighted sum of the inputs. The positive shadow prices in the primary are represented by positive lambda values in the dual model, which are used to identify inefficient units. This is true where the weighting structure is calculated through mathematical programming, and constant returns to scale (CRS) are assumed. In 1984, (Banker et al, 1984) developed a model focusing on variable returns to scale.

In our presidential-candidate site, we used several weight-reflecting, multi-attribute performance measures, such as sensitivity, specificity, bias, and variance of misclassification rate, defined as follows:

Sensitivity = Number of instances that represent a target word (1)

Specificity = Number of instances that represent a context word (2)

where $P(Y_F = y|x)$ is the probability that the outcome of an instance with input x is y

and $P(Y_H = y|x)$ is the probability that the outcome of an instance with input x is classified as y.

We used DEA to determine individual weights to assess input-oriented CRS efficiency, output slacks, and efficient output targets. Experience, education, and eligibility were made input classifiers, and elected and successful were output performance indicators. We then found the efficiency scores for each of the decision-making units. Using benchmarking, we determined inefficiencies and used linear programming to find the best weight for v_i to maximize the output.

$$\max h_o = \sum_{r=1}^{s} v_r y_{rm}$$

$$\text{s.t. } \sum^{s} v_r y_{rm} \leq 1 \quad for\ m = 1, \ldots, M \tag{3}$$

$r=1$

$$v_r = 0, s = 1,2,3,4$$

The highest efficiency allowed by the constraints leads to output weighting of the outputs. Therefore, $h_o^* = 1$, and the slack constraints were met if and only if word instance o was efficient relative to other instances considered. On the other hand, if $h_o^* < 1$, then the word instance was considered inefficient and therefore was not assigned a higher rating than the reference instance to which it was compared.

Results and Discussion

The use of this methodology led to the results below, including the input-oriented CRS efficiencies, output slacks, and efficient output targets, using experience, education, and eligibility as inputs; and elected and successful as outputs to determine the corresponding weights. The data used in this model are the results that collected, the words instances and corpus, from the web site which is one of the components of this research.

Collected Word Corpus

A dashboard developed in R using the Shiny package for data mining word instances and their contexts for the analysis of WSD in DEA. The first thing of exploring the content of the corpus is using some basic methods, typically by counting the words, such as finding out which words are the most frequent in the corpus, and in which subjects. Figure 10 gives an example of the collected corpus of word instances regarding the education, the environment, and the healthcare.

Fig. 10. Example of Collected Word Corpus

The inputs for these data were the candidate credentials regarding their experience, education, and eligibility. The outputs are related to them being elected and successful. Table 7 contains raw data of word instances gathered in the initial data collection.

Table 7. Number of appearances of key words from candidate surveys

Target word	Experience	Education	Eligibility	Elected	Successful
Good	17	60	10	1	7
Fair	51	1	42	1	5
Knowledge	55	1	31	1	8
Fit	42	14	30	2	11
Environment	50	2	2	2	48
Education	16	3	77	2	35

Terrorism	21	30	25	2	51
Health care	60	40	30	1	6
Inflation	1	100	21	1	1
Corrupt	2	9	91	1	10
Racism	37	66	33	2	47
R&D	55	51	99	2	88
Violence	5	74	16	1	12

Technical Efficiency

The DEA efficiency score is summarized to show the maximum output given minimum input. The results showed that the terms "good, fit, health care, racism and R&D" have scores of less than 1 but greater than 0, and thus they are identified as inefficient. Table 8 showed that the term good had an input-oriented CRS of 0.868 and an optimal lambda in the return to scale benchmark. Its increasing RTS lambda sum was 0.756, by virtue of the contribution of the environment (.244) and violence (.512). For its part, fit showed an input-oriented CRS of .807 and increasing RTS lambda sum of 1, resulting from environment (.478), education (.196), and terrorism (.325).

Table 8. Input-oriented CRS efficiency

DMU no.	**DMU name**	**CRS**	**Sum of Lambdas**	**RTS**	**Optimal lambdas**					
1	good	0.868	0.756	irs	0.244	environment	0.512	violence		
2	fair	1.000	0.500	crs	0.500	environment				
3	knowledge	1.000	0.500	crs	0.500	environment				
4	fit	0.807	1.000	irs	0.478	environment	0.196	education	0.325	terrorism
5	environment	1.000	1.000	crs	1.000	environment				
6	education	1.000	1.000	crs	1.000	education				
7	terrorism	1.000	1.000	crs	1.000	terrorism				
8	health care	0.268	0.511	irs	0.197	environment	0.293	terrorism	0.021	violence

9	inflation	1.000	1.000	crs	1.000	inflation				
10	corrupt	1.000	1.000	crs	1.000	corrupt				
11	racism	0.664	1.129	drs	0.173	environment	0.698	terrorism	0.258	violence
12	R&D	0.749	1.912	drs	0.135	environment	0.568	education	1.208	terrorism
13	violence	1.000	1.000	crs	1.000	violence				

DMU= decision making units, CRS = constant return to scale, RTS = return to scale, irs = increasing return to scale; drs = decreasing return to scale.

Likewise, healthcare had an input-oriented CRS efficiency of .268 and an increasing RTS, with a lambda sum of .511 that resulted from the contributions of environment (.197), terrorism (.293), and violence (.021). Racism and R&D had input-oriented CRS efficiencies of 0.665 and 0.750, respectively, and their lambdas were both over 1. The sum of lambdas for racism was 1.129, and the lambda sum for R&D was 1.912. These word instances showed decreasing RTS benchmarks. The remaining word instances for fair, knowledge, environment, education, terrorism, inflation, corrupt, and violence showed constant RTS benchmarks due to input-oriented CRS efficiencies of 1.00. However, among the inefficiencies terms, the term "good" is closer to an efficiency frontier, and needs 13.2 % reduction in inputs to be efficient. The details of these input reductions or output augmentations are called slacks.

Slacks and Targets

Table 9 presented the DEA slacks. In the classic radial models, the slacks relate to the increases in output or reductions in input that could be gained beyond what is implied by the radial projection. Therefore, the slacks are intended to investigate equal increases in all outputs or decreases in all inputs. Because we maximized the output of the instances of the words examined, efficient peers might use less of the input.

Table 9. Input-oriented CRS model slacks

DMU no.	DMU name	***Input slacks***			**Output slacks**	
		Experience	***Education***	***Eligibility***	**Elected**	**Successful**
1	good	0.00000	13.70549	0.00000	0.00000	10.8549
2	fair	26.0000	0.00000	41.0000	0.00000	19.0000
3	knowledge	30.0000	0.00000	30.0000	0.00000	16.0000
4	fit	0.00000	0.00000	0.00000	0.00000	35.4213
5	environment	0.00000	0.00000	0.00000	0.00000	0.00000

6	education	0.00000	0.00000	0.00000	0.00000	0.00000
7	terrorism	0.00000	0.00000	0.00000	0.00000	0.00000
8	health care	0.00000	0.00000	0.00000	0.00000	18.6258
9	inflation	0.00000	0.00000	0.00000	0.00000	0.00000
10	corrupt	0.00000	0.00000	0.00000	0.00000	0.00000
11	racism	0.00000	3.47974	0.00000	0.00000	0.00000
12	R&D	0.00000	0.00000	0.00000	1.82320	0.00000
13	violence	0.00000	0.00000	0.00000	0.00000	0.00000

In our sample, the instances of fit and health care showed no input slacks but did show successful output slacks of 35.421 and 18.626, respectively. The instance good had input slacks of 13.705 in education and 10.855 in successful outputs. Environment, education, terrorism, inflation, corruption, and violence had neither input nor output slacks, leading to a CRS of 1. Racism showed an input slack of 3.48 for education. Fairness and knowledge showed the greatest input and output slacks, which is consistent with their CRS. Fairness exhibited slack for experience (26) and eligibility (41), leading to a successful output slack of 19. Knowledge had an input slack of 30 in both experience and eligibility, with a successful output slack of 16. The data in Table 10 supported the conclusions from an input-oriented CRS model target perspective. To calculate the target values for inputs, the input value is multiplied with an optimal efficiency score, and then slack amounts are subtracted from this amount.

Table 10. Input-oriented CRS model target

		Efficient input target			**Efficient output target**	
DMU no.	**DMU name**	***Experience***	***Education***	***Eligibility***	**Elected**	**Successful**
1	good	14.75824	38.38242	8.68132	1.00000	17.85495
2	fair	25.00000	1.000000	1.00000	1.00000	24.00000
3	knowledge	25.00000	1.000000	1.00000	1.00000	24.00000
4	fit	33.89407	11.29802	24.2100	2.00000	46.42132
5	environment	50.00000	2.000000	2.00000	2.00000	48.00000
6	education	16.00000	3.000000	77.0000	2.00000	35.00000

7	terrorism	21.00000	30.00000	25.0000	2.00000	51.00000
8	health care	16.09266	10.72844	8.04633	1.00000	24.62580
9	inflation	1.000000	100.0000	21.0000	1.00000	1.000000
10	corrupt	2.000000	9.000000	91.0000	1.00000	10.00000
11	racism	24.58928	40.38223	21.9309	2.00000	47.00000
12	R&D	41.22674	38.22843	74.2081	3.82320	88.00000
13	violence	5.00000	74.00000	16.00000	1.00000	12.00000

Table 10 showed a plot of instances of elected against successful, which allows several patterns to emerge. R&D, racism, and terrorism appear much more prevalently where the focus is shifted from increasing returns to sheer numbers. This contrast is important for confirming the value of DEA. The WSD is not a product of numbers alone but rather of increasing returns to scale.

Predictive Neural Network

The results of the predictive neural network are given in Table 11. The issues of racism, R&D, and violence were removed from the model because they were used to train the neural network.

Table 11. Neural net prediction

Word	**Experience**	**Education**	**Eligibility**	**Elected**	**Successful**	**$S-successful**
Good	17	60	10	1	7	7
Fair	51	1	42	1	5	6.951
Knowledge	55	1	31	1	8	6.75
Fit	42	14	30	2	11	47.434
Environment	50	2	2	2	48	47.672
Education	16	3	77	2	35	46.857
Terrorism	21	30	25	2	51	47.104
Health care	60	40	30	1	6	6.217

Inflation	1	100	21	1	1	7.081
Corrupt	2	9	91	1	10	8.653

The results demonstrate the validity of using DEA as a tool for WSD in that it uses the entire sample and does not require part of the population to be split off as a training sample. DEA operates on a frontier delimited by constant, decreasing, and increasing RTS. It is not dependent on sheer numbers alone.

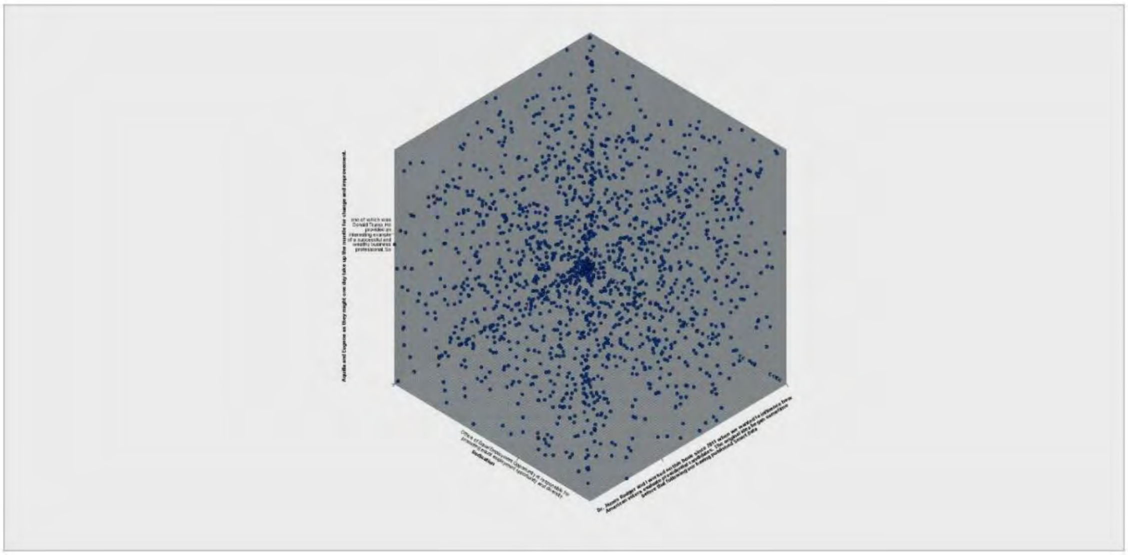

Fig. 11. Original Word Instance Visualization

Figure 11 presents a visualization of the original word instances that were text mined for inputting into the DEA analysis. This indicates the thirteen instances analyzed for increasing RTS impact by using Shiny on the results of the responses to the questionnaires.

Conclusions

Based on the theories of word sense disambiguation and ontologies, we have referenced the literature and come to the following conclusions (Wang et al, 2014), (Huang et al, 2015), (Lastra-Díaz et al, 2019). Our results indicate that the survey administered by the website used in this study is a promising tool for predicting successful presidential candidates. The initial survey was completed by middle class millennials, and it seems clear that voters of this generation would perceive attention to environmental issues and freedom from violence to be 'good.' Likewise, the respondents perceived candidates to be fit in relation to responses to the environment, education, and terrorism. Finally, these voters disambiguated health care from the concept of freedom from the violence and terrorism that has become so prevalent an issue in their lives and the dominant media narratives. Such voters do not appear to see issues of racism and R&D as nearly as important. Further, they consider violence, corruption, inflation, terrorism, education, environment, knowledge, and fairness separate entities unto themselves. For this convenience sample of the younger population, a good, fit, and caring candidate will attend to their concerns for the environment, education, violence, and terrorism, which are the benchmarks in their lives. Future investigation of additional respondents collected through our site will examine a more diverse and nationalized sample to determine how instances of WSD might vary with age, income, and other descriptive statistics.

References

Abualhaija & Zimmermann. (2016). D-Bees: A novel method inspired by bee colony optimization for solving word sense disambiguation. *Swarm and Evolutionary Computation: Elsevier*, Volume 27, April 2016, Pages 188-195: https://doi.org/10.1016/j.swevo.2015.12.002.

Akono et al. (2013). Institutions and Hospital Efficiency in Cameroon: A Data Envelopment Analysis. *Journal of African Development : Spring*, 45-71.

AlMaayah et al. (2016). Towards an automatic extraction of synonyms for Quranic Arabic WordNet. *International Journal of Speech Technology*, Volume 19, Issue 2, pp 177–189. https://doi.org/10.1007/s10772-015-9301-9.

Amy Neustein. (2008). Note from the editor: “Rethinking natural language processing for speech technology”. *Int J Speech Technol: Springer Science+Business Media, LLC 2009*, 11: 107–108: DOI 10.1007/s10772-009-9057-1.

Amy Neustein. (2012). *Think before You Talk: The Role of Cognitive Science in Natural Language Processing.* USA: Linguistic Technology Systems, 800 Palisade Avenue, Suite 1809, Fort Lee, NJ 07024 U.S.A: http://www.amyneustein.com/pubs/neustein.nlpcs.2012.paper.pdf.

Andres Duque et al. (2018). Co-occurrence graphs for word sense disambiguation in the biomedical domain. *Artificial Intelligence in Medicine: Elsevier*, Volume 87, May 2018, Pages 9-19: https://doi.org/10.1016/j.artmed.2018.03.002.

Antonio Yepes. (2017). Word embeddings and recurrent neural networks based on Long-Short Term Memory nodes in supervised biomedical word sense disambiguation. *Journal of Biomedical Informatics: Elsevier*, Volume 73, September 2017, Pages 137-147: https://doi.org/10.1016/j.jbi.2017.08.001.

Aparicioa et al. (2019). Measuring efficiency in education: The influence of imprecision and variability in data on DEA estimates. *Socio-EconomicPlanningSciences: Elsevier Ltd. All rights reserved.*, 1-12: https://doi.org/10.1016/j.seps.2019.03.004.

Arroyo-Fernández et al. (2019). Language features in extractive summarization: Humans Vs. Machines. *Knowledge-Based Systems*, Volume 180, 15 September 2019, Pages 1-11: https://doi.org/10.1016/j.knosys.2019.05.014.

Banker et al. (1984). Some Models for Estimating Technical and. *Management Science*, 30(9), 1078-1092: doi:10.1287/mnsc.30.9.1078.

Bateman. (2010). Language and Space: a two-level semantic approach based on principles of ontological engineering. *International Journal of Speech Technology: Springer*, Volume 13, Issue 1, pp 29–48: https://doi.org/10.1007/s10772-010-9069-x.

Bhatia et al. (2016). Ontologies for Software Engineering: Past, Present and Future. *Indian Journal of Science and Technology*, Vol 9(9), DOI: 10.17485/ijst/2016/v9i9/71384.

Bhattacharyya et al. (2018). *Handbook of signal processing systems: Third Edition.* Belgium: Springer International Publishing AG, part of Springer Nature 2019: https://doi.org/10.1007/978-3-319-91734-4_12.

Bork et al. (2019). A survey of modeling language specification techniques. *Information Systems: Elsevier*, Volume 87, January 2020, 101425: https://doi.org/10.1016/j.is.2019.101425.

Boudchiche & Mazroui. (2019). A hybrid approach for Arabic lemmatization. *International Journal of Speech Technology Springer*, September 2019, Volume 22, Issue 3, pp 563–573. https://doi.org/10.1007/s10772-018-9528-3.

Cetrulo et al. (2019). An analytical review of the efficiency of water and sanitation utilities in developing countries. *Water Research: Elsevier Ltd.*, 372-380: https://doi.org/10.1016/j.watres.2019.05.044.

Charnes et al. (1978). Measuring the Efficiency of Decision Making Units. *European Journal of Operational Research: Elsevier*, Volume 2, Issue 6, Pages 429–444: https://doi.org/10.1016/0377-2217(78)90138-8.

Chatzigeorgiou & Stiakakis. (2010). Benchmarking library and application software with Data Envelopment Analysis. *Software Qual: Springer Science+Business Media*, 1-26: https://doi.org/10.1007/s11219-010-9113-8.

Chengyao Lv et al. (2016). Corpus based part-of-speech tagging. *International Journal of Speech Technology: Springer*, 19(3): 10.1007/s10772-016-9356-2.

Chifu et al. (2015). Word sense discrimination in information retrieval: A spectral clustering-based approach. *Information Processing and Management: Elsevier*, Volume 51, Issue 2, March 2015, Pages 16-31: https://doi.org/10.1016/j.ipm.2014.10.007.

Cooper et al. (2011). *Data Envelopment Analysis: History, Models and Interpretations.* Boston: Springer.

CorrêaJr et al. (2018). Word sense disambiguation: A complex network approach. *Information Sciences: Elsevier*, Volumes 442–443, May 2018, Pages 103-113: https://doi.org/10.1016/j.ins.2018.02.047.

Dan Cristea. (2009). Motivations and implications of veins theory: a discussion of discourse cohesion. *International Journal of Speech Technology: Springer*, 12: 83. https://doi.org/10.1007/s10772-009-9059-z.

Donald Bell. (2004). The sequence diagram. *IBM Developer*, https://developer.ibm.com/articles/the-sequence-diagram/.

Duque et al. (2016). Can multilinguality improve Biomedical Word Sense Disambiguation? *Journal of Biomedical Informatics: Elsevier*, Volume 64, December 2016, Pages 320-332: https://doi.org/10.1016/j.jbi.2016.10.020.

Duque et al. (2018). Co-occurrence graphs for word sense disambiguation in the biomedical domain. *Artificial Intelligence in Medicine: Elsevier*, Volume 87, May 2018, Pages 9-19: https://doi.org/10.1016/j.artmed.2018.03.002.

El Mahdaouy et al. (2018). Improving Arabic information retrieval using word embedding similarities. *International Journal of Speech Technology: Springer*, Volume 21, Issue 1, pp 121–136: https://doi.org/10.1007/s10772-018-9492-y.

Emrouznejad & Shale. (2008). A combined neural network and DEA for measuring efficiency of large scale datasets. *Computers & Industrial Engineering: Elsevier*, 56 (2009) 249–254: doi:10.1016/j.cie.2008.05.012.

Frank Xing et al. (2019). Sentiment-aware volatility forecasting. *Knowledge-Based Systems : Elsevier*, Volume 176, 15 July 2019, Pages 68-76: https://doi.org/10.1016/j.knosys.2019.03.029.

George & Rodger. (2017). *How to Select an American President.* Bloomington: ArchwayPublishing: https://www.archwaypublishing.com: ISBN-10: 1480840882, ISBN-13: 978-1480840881.

Gosálbez et al. (2019). Assessing the performance of UK universities in the field of chemical engineering using data envelopment analysis. *Education for Chemical Engineers: Elsevier*, Volume 29, Pages 29-41:https://doi.org/10.1016/j.ece.2019.06.003.

Henriques et al. (2018). Efficiency in the Brazilian banking system using data envelopment analysis. *Future Business Journal: Elsevier*, 157-178: https://doi.org/10.1016/j.fbj.2018.05.001.

Huang et al. (2015). Unsupervised word sense induction using rival penalized competitive learning. *Engineering Applications of Artificial Intelligence: Elsevier*, Volume 41, May 2015, Pages 166-174: https://doi.org/10.1016/j.engappai.2015.02.004.

Hung & Chen. (2016). Word sense disambiguation based sentiment lexicons for sentiment classification. *Knowledge-Based Systems: Elsevier*, Volume 110, 15 October 2016, Pages 224-232: https://doi.org/10.1016/j.knosys.2016.07.030.

Ion & Tufiş. (2009). Multilingual versus monolingual word sense disambiguation. *International Journal of Speech Technology : Springer*, 12: 113. https://doi.org/10.1007/s10772-009-9053-5.

ItI Education. (2010). *Introduction to Database Systems.* Delhi: Pearson Education India.

Izquierdo & Su ´ arez. (2009). An Empirical Study on Class-based Word Sense Disambiguation. *Proceedings of the 12th Conference of the European Chapter of the ACL: Association for Computational Linguistics*, pages 389–397.

Jablonský & Josef. (2016). Ranking models in data envelopment analysis. *Business Trends. Trendy v podnikání*, vol. 6, no. 4, p. 36-42, 2016.

Jianping Yu et al. (2016). A new approach of attribute partial order structure diagram for word sense disambiguation of English prepositions. *Knowledge-Based Systems: Elsevier*, Volume 95, 1 March 2016, Pages 142-152: https://doi.org/10.1016/j.knosys.2015.12.014.

Jie Xu et al. (2019). Visual-textual sentiment classification with bi-directional multi-level attention networks. *Knowledge-Based Systems: Elsevier*, Volume 178, 15 August 2019, Pages 61-73: https://doi.org/10.1016/j.knosys.2019.04.018.

Kadim & Lazrek . (2016). Bidirectional HMM-based Arabic POS tagging. *International Journal of Speech Technology*, Volume 19, Issue 2, pp 303–312: https://doi.org/10.1007/s10772-015-9303-7.

Krawczyk & Mcinnes. (2018). Local Ensemble Learning from Imbalanced and Noisy Data for Word Sense Disambiguation. *Pattern Recognition: Elsevier*, 78:103-119: DOI: 10.1016/j.patcog.2017.10.028.

Lakhfif & Laskri. (2016). A frame-based approach for capturing semantics from Arabic text for text-to-sign language MT. *International Journal of Speech Technology*, Volume 19, Issue 2, pp 203–228: https://doi.org/10.1007/s10772-015-9290-8.

Laorden et al. (2012). Word sense disambiguation for spam filtering. *Electronic Commerce Research and Applications: Elsevier*, Volume 11, Issue 3, May–June 2012, Pages 290-298: https://doi.org/10.1016/j.elerap.2011.11.004.

Lastra-Díaz et al. (2019). A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art. *Engineering Applications of Artificial Intelligence: Elsevier*, Volume 85, October 2019, Pages 645-665: https://doi.org/10.1016/j.engappai.2019.07.010.

Lopez-Arevalo et al. (2017). Improving selection of synsets from WordNet for domain-specific word sense disambiguation. *Computer Speech & Language: Elsevier*, Volume 41, January 2017, Pages 128-145: https://doi.org/10.1016/j.csl.2016.06.003.

Marwah Alian et al. (2016). Word sense disambiguation for Arabic text using Wikipedia and Vector Space Model. *International Journal of Speech Technology*, 19 DOI: 10.1007/s10772-016-9376-y.

Menai. (2014). Word sense disambiguation using evolutionary algorithms – Application to Arabic language. *Computers in Human Behavior: Elsevier*, Volume 41, December 2014, Pages 92-103: https://doi.org/10.1016/j.chb.2014.06.021.

Murrar et al. (2017). The Efficiency and Institutional Performance of the Palestinian Water Service Providers. *American Journal of Environmental and Resource Economics*, 162-174: doi: 10.11648/j.ajere.20170204.13.

Oi YeeKwong. (2008). Sense abstractness, semantic activation, and word sense disambiguation. *International Journal of Speech Technology: Springer*, December 2008, 11:135: https://doi.org/10.1007/s10772-009-9041-9.

Scheel et al. (2015). The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM, Volume I :Phase 2: Process Concept Evolution. *Elsevier*, 2015, Pages 11-35: https://doi.org/10.1016/B978-0-12-799959-3.00002-1.

Sestayo & Castro. (2018). The impact of tourist destination on hotel efficiency: A data envelopment analysis approach. *European Journal of Operational Research: Elsevier B.V. All rights reserved.*, 674-686: https://doi.org/10.1016/j.ejor.2018.06.043.

Siafakas et al. (2019). Identification of efficient dairy farms in Greece based on home grown feed stuffs, using the Data Envelopment Analysis method. *Livestock Science: Elsevier B.V. All rights reserved*, 14-20: https://doi.org/10.1016/j.livsci.2019.02.008.

Silke Maren Witt. (2015). Modeling user response timings in spoken dialog systems. *International Journal of Speech Technology: Springer*, 18(2): DOI: 10.1007/s10772-014-9265-1.

Singh & Gupta. (2019). A novel unsupervised corpus-based stemming technique using lexicon and corpus statistics. *Knowledge-Based Systems: Elsevier*, Volume 180, 15 September 2019, Pages 147-162: https://doi.org/10.1016/j.knosys.2019.05.025.

Singha & Siddiquia . (2015). Role of Semantic Relations in Hindi Word Sense Disambiguation. *Procedia Computer Science: Elsevier*, Volume 46, 2015, Pages 240-248: https://doi.org/10.1016/j.procs.2015.02.017.

Tinghua Wang et al. (2017). Sprinkled semantic diffusion kernel for word sense disambiguation. *Engineering Applications of Artificial Intelligence: Elsevier*, Volume 64, September 2017, Pages 43-51: https://doi.org/10.1016/j.engappai.2017.05.010.

Wang et al. (2013). China's regional energy and environmental efficiency: A DEA window analysis based dynamic evaluation. *Mathematical and Computer Modelling: Elsevier*, Volume 58, Issues 5–6, September 2013, Pages 1117-1127: https://doi.org/10.1016/j.mcm.2011.11.067.

Wang et al. (2014). Supervised word sense disambiguation using semantic diffusion kernel. *Engineering Applications of Artificial Intelligence: Elsevier*, Volume 27, January 2014, Pages 167-174: https://doi.org/10.1016/j.engappai.2013.08.007.

Yoan Gutiérrez et al. (2017). Spreading semantic information by Word Sense Disambiguation. *Knowledge-Based Systems: Elsevier*, Volume 132, 15 September 2017, Pages 47-61: https://doi.org/10.1016/j.knosys.2017.06.013.

Zhou et al. (2019). Haze Influencing Factors: A Data Envelopment Analysis Approach. *International Journal of Environmental Research and Public Health: MDPI*, 1-16: doi: 10.3390/ijerph16060914.