

# Analysis of Natural Language Processing (NLP) approaches to determine semantic similarity between texts in domain-specific context



Utrecht University

Author: Surabhi Som (6248160)  
[s.som@students.uu.nl](mailto:s.som@students.uu.nl)

Supervisors: Denis Paperno  
[d.paperno@uu.nl](mailto:d.paperno@uu.nl)

Rick Nouwen  
[r.w.f.nouwen@uu.nl](mailto:r.w.f.nouwen@uu.nl)

*A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Artificial Intelligence*

in

**Faculty of Science  
Utrecht University**

# Table of Contents

|  |           |
|--|-----------|
| <b>Chapter 1</b> .....                                       | <b>3</b>  |
| <b>Introduction</b> .....                                    | <b>3</b>  |
| <b>1.1 Problem Description</b> .....                         | <b>4</b>  |
| <b>Chapter 2</b> .....                                       | <b>5</b>  |
| <b>Literature Review</b> .....                               | <b>5</b>  |
| 2.1 Natural Language Processing .....                        | 5         |
| 2.2 Ontologies and its importance .....                      | 5         |
| 2.3 Semantics .....  | 6         |
| 2.4 Semantic Similarity.....                                 | 7         |
| 2.5 Sentence Semantic Similarity .....                       | 15        |
| 2.6 Text Semantics.....                                      | 23        |
| 2.7 Stemming and Lemmatization .....                         | 25        |
| <b>Chapter 3</b> .....                                       | <b>33</b> |
| <b>Research Methodology</b> .....                            | <b>33</b> |
| <b>Chapter 4</b> .....                                       | <b>38</b> |
| <b>Data Collection and Analysis</b> .....                    | <b>38</b> |
| <b>Chapter 5</b> .....                                       | <b>44</b> |
| <b>Results</b> .....   | <b>44</b> |
| <b>Chapter 6</b> .....                                       | <b>46</b> |
| <b>Conclusion</b> .....                                      | <b>46</b> |
| <b>List of References</b> .....                              | <b>47</b> |
| <b>Appendix</b> .....  | <b>57</b> |
| <b>Appendix A: Gold Standard Data</b> .....                  | <b>57</b> |
| <b>Appendix B: Yes/No Questionnaire</b> .....                | <b>60</b> |
| <b>Appendix C: Individual Clustering Questionnaire</b> ..... | <b>64</b> |
| <b>Appendix D: Sentence Agreement Program</b> .....          | <b>65</b> |

# Chapter 1

## Introduction

Natural Language Processing (NLP) is a discipline which makes an effort to “gather knowledge on how human beings understand and use language so that appropriate tools and techniques can be developed to make computer systems understand and manipulate natural languages to perform the desired tasks” Chowdhury (2003, p.51). This would enable better human-computer interaction and communication between human and computer will improve further.

Liddy (2001; p.2) defines Natural Language Processing (NLP) as follows:

*“Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications.”*

The major contributors to the development and applications of NLP are Linguistics, Mathematics, Computer Science, Electrical and Electronic Engineering, Artificial Intelligence, Robotics, and Cognitive Psychology (Liddy 2001; Chowdhury 2003). NLP has emerged as an important discipline because of its diverse applications in mobiles, retail business, education, healthcare (Zhou, et al., 2006), defence (Hancox<sup>1</sup>) and various other areas which are crucial to our everyday life. Some of the common applications of NLP include machine translation, user interfaces, multilingual and cross language information retrieval (Chowdhury, 2003), text classification and categorization (such as web searching, information filtering, language identification, readability assessment, and sentiment analysis), named entity recognition (classifying named entities, into predefined categories like persons, organizations, locations, time, dates, etc.), part-of-speech tagging (parsing, text-to-speech conversion, information extraction, etc.), semantic parsing and question answering (automatically answer different types of questions asked in natural languages including definition questions, biographical questions, multilingual questions, etc.) paraphrase detection, natural language generation (such as automated writing of reports based on data analysis in retail business, medical records, etc.), speech recognition (home automation, mobile telephony, virtual assistance, hands-free computing, video games, etc.), character recognition and spell checking<sup>2</sup>.

Major business applications of NLP emerge from the fact that Artificial Intelligence is assisting businesses to efficiently handle various core business issues. Hence NLP applications are in customer service (using speech recognition & question answering), reputation monitoring (using sentiment analysis and co-reference resolution), advertisement (using keyword matching and sense disambiguation), market intelligence (using event extraction and sentence classification) and regulatory compliance (using named entity recognition and relation detection)<sup>3</sup>. Machine translation, sequence modelling, named entity recognition, part-of-speech tagging, sentiment analysis are some of the routine tasks in natural language processing that work on the basis of question answering problems over

<sup>1</sup> [https://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1\\_history.html](https://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1_history.html)

<sup>2</sup> <https://medium.com/@datamonsters/artificial-neural-networks-in-natural-language-processing-bcf62aa9151a>

<sup>3</sup> <https://emerj.com/ai-sector-overviews/natural-language-processing-business-applications/>

language input (Kumar et al, 2016). Both for internal (e.g. human resource operations) and external (e.g. customer service) projects, similar technology is being used by companies<sup>4</sup>.

However, currently there are various limitations to the NLP systems. In order to make a quantum jump from Natural Language Processing to Natural Language Understanding, the research needs to focus on semantically related concepts that will enable the performance of complex NLP tasks, just like us, the human text processors which can ‘see more than what we see’ (Cambria & White, 2014). This research focuses on the semantic similarity between texts in domain-specific context, with the attempt to make a contribution towards paradigm shift from Natural Language Processing to Natural Language Understanding.

## 1.1 Problem Description

Research in NLP has focused their efforts on various tasks. Natural Language Understanding (NLU), which maps text to its meaning, is working on interpretation of text. Since there is a lack of studies that integrate the different branches of research to incorporate text semantics in the text mining process, secondary studies, such as surveys and reviews, can integrate and organize the studies that were already developed to guide future research in related areas (Sinoara *et al.* 2017). Semantic similarity research has so far been carried out to find out solutions to various domains. However, it has still not been applied to human resources (HR) domain. In my research, I want to focus on this specific domain and hence the following research questions will be the main focus areas:

**RQ1:** What is the level of agreement between sentence pair classification based on semantic similarity between human annotated gold standard data and questionnaire data?

**RQ2:** What can be an alternative method for collection of gold standard data for measuring semantic similarity?

The above questions answered will aid in the creation of gold standard data which can be used as a corpus for input to the various semantic similarity tasks in HR domain.

<sup>4</sup> <https://emerj.com/ai-sector-overviews/natural-language-processing-business-applications/>

# Chapter 2

## Literature Review

### 2.1 Natural Language Processing

Natural Language Processing (NLP) is fundamental to artificial intelligence for communicating with intelligent systems using natural languages. NLP assists computers to understand, analyze, and derive meaning from everyday human language in a smart and useful way (Lu *et al.*, 2018) Examples of natural language processing systems in artificial intelligence in everyday life include better communication (Facebook Messenger using artificial intelligence, Skype Translator), faster clinical diagnosis, customer review and intelligent personal assistants (IPAs).<sup>6</sup> IPAs (such as Apple's Siri, Google Now and Microsoft Cortana) are programmed within Artificial Intelligence (AI) do create an interaction between human and computer through a natural language used in digital communication (Canbek & Mutlu, 2016). Sil et al. (2010, p.1) argue that Artificial Intelligence (AI) researchers have to recognize that to develop common sense knowledge in machines for information extraction, "the dynamics of the world is arguably the most crucial form of knowledge for an intelligent agent, since it informs an agent of the ways in which it can act upon the world".

### 2.2 Ontologies and its importance

Ontologies have been used in Artificial Intelligence "to develop an understandable, complete, and sharable system of categories, labels, and relationships that represent the real world in an objective manner" (Kim & Storey 2011, p.2). Ontology is a hierarchical catalogue of the concepts that a person has in mind, where the semantic knowledge is stored in the form of meaning postulates (Periñán-Pascual & Arcas-Túnez, 2010). In the context of artificial intelligence, it "includes machine-interpretable definitions of basic concepts in the domain and relations among them"<sup>7</sup>. While referring to the world wide web, ontology would represent "a set of concepts and the relationships among them for a particular domain" (Kim & Storey 2011, p.2). Uschold & Gruninger (1996) conceptualized ontology "to refer to the shared understanding of some domain of interest which may be used as a unifying framework which entails taking a world view" (conceived as a set of concepts entities, attributes, processes, their definitions and their interrelationships) with respect to a given domain. Ontologies represent a schema for a particular domain, tend to provide expert background knowledge about a domain by clarifying technical terms or specifying relationship between concepts (Stavrianou et al., 2007; p.31). Ontologies provide a formal specification of a shared conceptualization constructed from the consensus of a community of users or domain experts and they represent a very reliable and structured knowledge source, which is machine readable (Sánchez et al. 2012; p.7719). However, there are a number of challenges to developing ontologies, which include they are specific to each domain and are time-consuming to create and creating large-scale ontologies such as Cyc require a collaborative, community effort from knowledgeable people (Kim & Storey 2011, p.2).

<sup>5</sup> <https://www.expertsystem.com/examples-natural-language-processing-systems-artificial-intelligence/>

<sup>6</sup> <https://www.expertsystem.com/examples-natural-language-processing-systems-artificial-intelligence/>

<sup>7</sup> [https://protege.stanford.edu/publications/ontology\\_development/ontology101-noy-mcguinness.html](https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html)

Ontology includes a vocabulary of terms and some specification of their meaning but the degree of formality by which a vocabulary is created and meaning is specified varies considerably is a continuum: ranging from (1) highly informal - expressed loosely in natural language (2) semi-informal - expressed in a restricted and structured form of natural language greatly increasing clarity and reducing ambiguity (3) semiformal - expressed in an formally defined language (4) rigorously formal - meticulously defined terms with formal semantics, theorems and proofs of such properties as soundness and completeness (Uschold & Gruninger,1996; p.6).

Kim & Storey (2011, p.2) state that “domain ontologies specify concepts, relationship among concepts, and inference rules for a single application domain (e.g., airline reservations, art galleries, furniture, fishing, gourmet food) or task”. A proper encoding mechanism is used in ontology, to formulate concepts in a specific domain that can support efficient information retrieval and reduced information overload while dealing with huge sets of data (Vairavasundaram & Logesh, 2018). Developing ontology helps to share common understanding of the structure of information among people or software agents, enables reuse of domain knowledge, makes domain assumptions explicit, separates domain knowledge from the operational knowledge and analyze domain knowledges. While extracting information from various text sources, “ontologies have been proposed for handling semantic heterogeneity” (Stavrianou et al., 2007; p.31). Ontologies have been extensively exploited in knowledge-based methods measures to compute semantic similarity (Sánchez et al. 2012; p.7719).

Adopting a corpus based approach, Vairavasundaram & Logesh (2018) developed an automatic topic ontology construction process (relying the on concept acquisition and semantic relation extraction) for better topic classification to enrich the set of categories in the Open Directory Project (ODP is a multilingual open content directory of World Wide Web links) by automatically identifying concepts and their associated semantic relationships based on external knowledge from Wikipedia and WordNet. They used a semantic similarity clustering algorithm to compute similarity and semantic relation extraction algorithm derived associated semantic relations between the set of extracted topics from the lexical patterns in WordNet. When evaluated for the classification of web documents, the performance of topic ontology was better over ODP.

Shift in NLP research has led to applications of statistical methods (such as machine learning and data mining) that have opened up fascinating areas of applications of traditional artificial intelligence techniques (Aggarwal, 2011), but one of the fundamental issues in artificial intelligence research (attempting to equip machines with the ability to understand natural language) is “how to represent language semantics in a way that can be manipulated by computers” (Gabrilovich & Markovitch, 2009; p. 443).

## 2.3 Semantics

Semantics focuses on the intrinsic meaning associated with natural language text (Cambria & White, 2014), takes into account the insightful understanding of the entities and it is at the underlying concept of numerous NLP applications (Harispe *et al.*, 2015). For example, text semantics is used for many information retrieval tasks such as search and text categorization

8 [https://protege.stanford.edu/publications/ontology\\_development/ontology101-noy-mcguinness.html](https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html)

(Sebastiani, 2002). Cambria & White (2014) argue that semantics-based approaches rely on implicit denotative features and are able to detect relevant information conveyed in a subtle manner. However, critics such as Gabrilovich & Markovitch (2009 p. 443) express the opinion that “this simple model can only be reasonably used when texts are fairly long, and performs sub-optimally on short texts”. Cambria & Hussain, 2012, pp.20-21) suggests that semantics could be better characterized by a “bag-of-concepts model”.

Semantics-based NLP approaches can be broadly classified into two main categories: Endogenous NLP that encompasses the usage of machine-learning methods to implement semantic analysis of a corpus by making structures that estimate notions from a large set of documents and Taxonomic NLP comprises initiatives that intend to create universal taxonomies or web ontologies for grasping the hierarchical semantics connected with natural language expressions. (Cambria & White, 2014, p.53).

## 2.4 Semantic Similarity

Similarity based on meaning as opposed to form is referred to as semantic similarity and people draw inductive inferences are drawn by people on the basis of semantic similarity (Hahn & Heit, 2015). We could define semantic similarity as a measure of how close are the semantic representations of different entities (such as units of language, e.g. words, sentences, or concepts) in a given knowledge base (Couto & Lamurias, 2019). Semantic similarity plays a vital role in the field of artificial intelligence, natural language processing, data mining and data processing and is useful in information management systems, especially when data from different various sources are to be collated in a meaningful manner (Gupta et al. 2017). Sánchez & colleagues (2012, p.7726) point out that “semantic similarity assessment is a crucial component embedded in many applications framed in the artificial intelligence research”. Some of the important applications of semantic similarity include time series analysis, information retrieval, finding near duplicate web pages, collaborative filtering, caching and audio files (Chauhan & Batra, 2018, p.714).

Most of the research work in the area of semantic similarity give emphasise to word-to-word similarity metrics and focused on the applications of the “traditional vectorial model, occasionally extended to n-gram language models” probably because of the easy “availability of resources that specifically encode relations between words or concepts (e.g. WordNet), and the various test beds that allow for their evaluation (e.g. TOEFL or SAT analogy/synonymy tests)” (Mihalcea *et al.*, 2006; p.776).

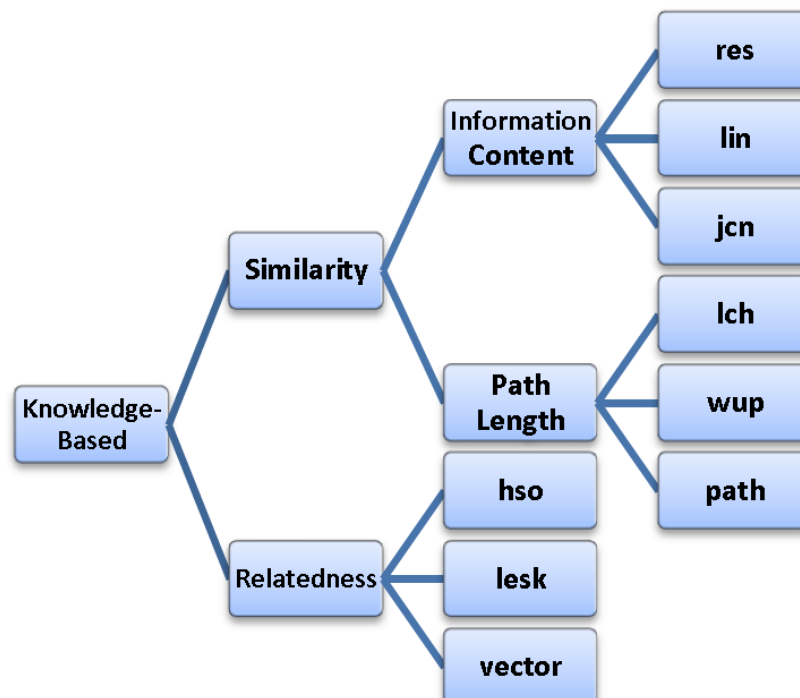
Semantic similarity is a measure that is used to compute the similarity between two concepts within ontology (Banu, 2015). Presently, semantic similarity methods have diverse usage for comparing primary elements of language, concepts, instances or even resources indexed by them (Harispe *et al.*, 2015). The two common methods for computing semantic similarity of two words are: dictionary-based methods (Inkpen, 2007) or knowledge based methods (Gupta *at el.*, 2017) (using WordNet, Roget’s Thesaurus, etc.) where short path means a high similarity and corpus-based methods (co-occurrence frequencies in corpora e.g. British National Corpus (BNC), TREC data, Waterloo Multitext, LDC English Gigabyte corpus, etc.) whereas the hybrid methods combine the two types (Inkpen, 2007, p.12-13). Corpus-based semantic similarity measure tries to identify the degree of similarity between words according to information derived from large corpora whereas knowledge-based semantic similarity try to identify the degree of similarity between words using information derived from semantic networks (Mihalcea *et al.* 2006).

Lin (1998) argues that a problem with some of the semantic similarity measures has been that “each of them is tied to a particular application or assumes a particular domain model and their underlying assumptions are often not explicitly stated, which makes it impossible to make theoretical arguments for or against any such measures”. A universal definition of similarity is proposed by Lin (1998) below is derived from a set of assumptions about similarity (because the author believes that if the assumptions are deemed reasonable, the similarity measure necessarily follow:

“Since similarity is the ratio between the amount of information in the commonality and the amount of information in the description of the two objects, if we know the commonality of the two objects, their similarity tells us how much more information is needed to determine what these two objects are”.

Researchers working on semantic similarity are showing increasing interest in ontologies because “they offer a structured and unambiguous representation of knowledge in the form of conceptualizations interconnected by means of semantic pointers” (Sánchez et al.2012). Perrián-Pascual & Arcas-Túnez (2007) state that “semantic knowledge is represented in the form of meaning postulates in the ontology”. Ontology-based similarity measures have been developed based on the similarity computation principle and depending on the way ontology is exploited (Sánchez et al.2012).

Ontology, taxonomies and semantic net are the knowledge representation forms which are used in information retrieval and these knowledge representations are used by various methods to find semantic similarity between different terms or concepts (Gupta et al, 2017). Knowledge based measures of semantic similarity are described below in Figure 1.

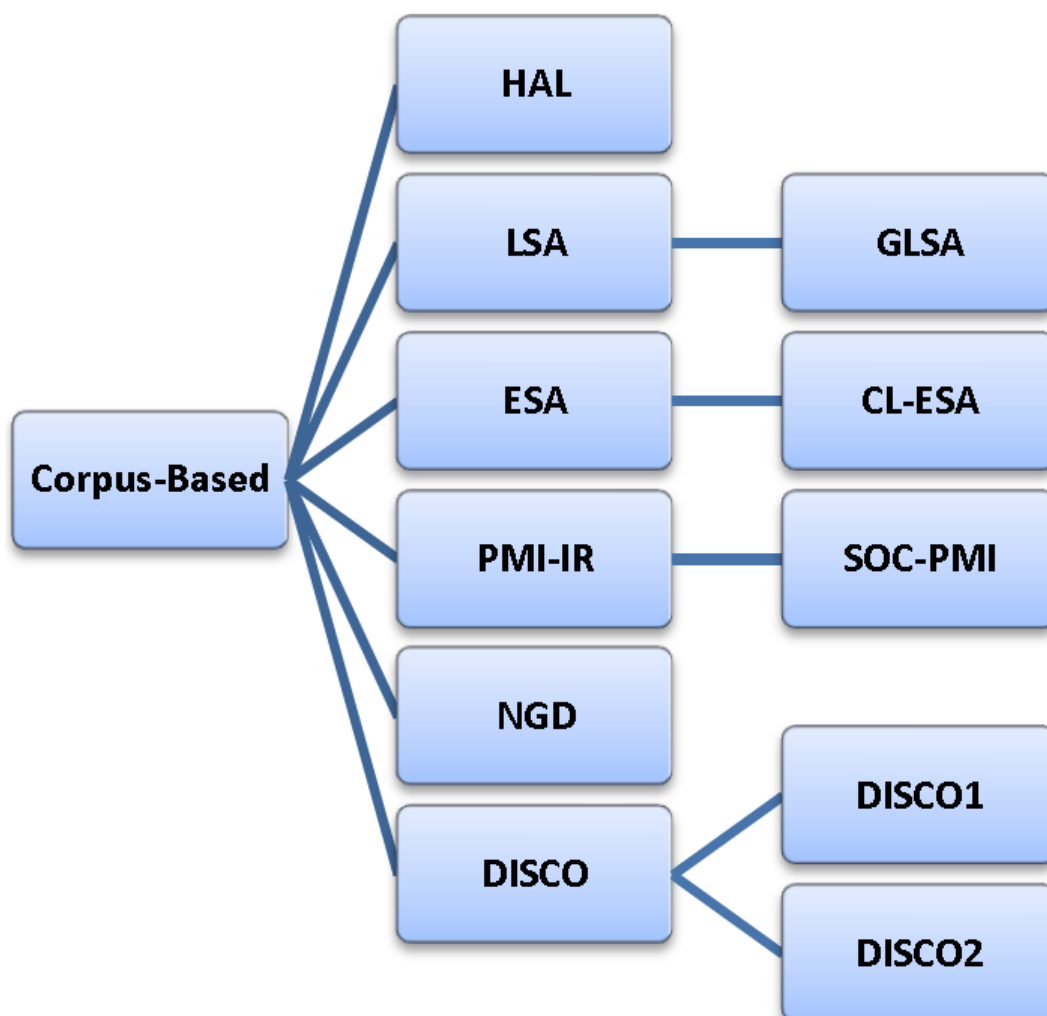


**Figure-1: Knowledge-based measures of semantic similarity (Gomaa & Fahmy, 2013)**

[Abbreviations in Figure-1 explained: St.Onge (hso), Lesk (lesk), vector pairs (vector), Resnik (res), Lin (lin), Jiang & Conrath (jcn), Leacock & Chodorow (lch), Wu & Palmer (wup) and Path Length (path)].



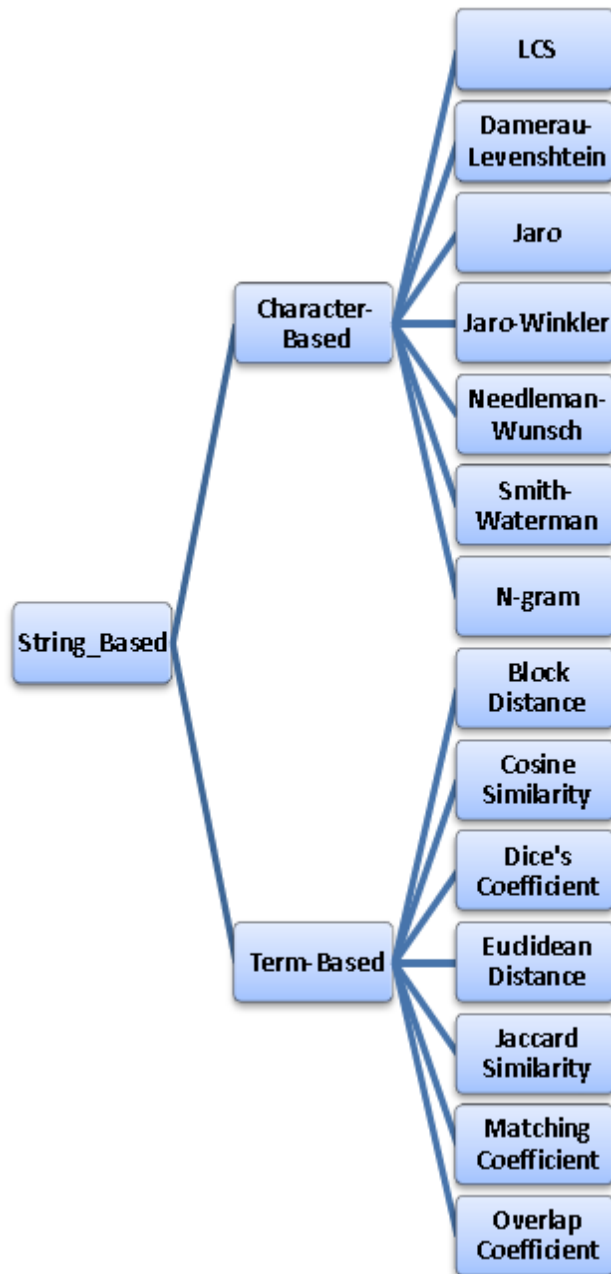
Corpus refers to a large collection of written material and speeches that are used to study and describe a language and such data available online can be used in determining the semantic relatedness among different words or concepts (Gupta *et al.*,2017). The various corpus-based similarity measures are summarized below in Figure-2.



**Figure-2: Corpus-based measures of semantic similarity** (Gomaa & Fahmy, 2013)

[Abbreviations used in Figure-2 explained: Hyperspace Analogue to Language (HAL), Latent Semantic Analysis (LSA), Generalized Latent Semantic Analysis (GLSA), Explicit Semantic Analysis (ESA), Cross-language explicit semantic analysis (CL-ESA), Pointwise Mutual Information - Information Retrieval (PMI-IR), Second-order co-occurrence pointwise mutual information (SCO-PMI), Normalized Google Distance (NGD), Extracting DIStributively similar words using CO-occurrences (DISCO)]

Gomaa & Fahmy (2013), argue that there are third type String-based measures, which “operate on string sequences and character composition, and a string metric is a metric that measures similarity or dissimilarity (distance) between two text strings for approximate string matching or comparison”. Further String-based measures are divided into two types character-based and term-based measures. String-based measures of semantic similarity are described below in Figure-3.



**Figure-3: String-based measures of semantic similarity** (Gomaa & Fahmy, 2013)

[Abbreviations in Figure-3 explained: Longest Common Sub-String (LCS) algorithm]

Cosine, Jaccard and Dice are some of the best-known techniques and most popular methods for finding the similarity that have been applied successfully in information retrieval systems (Strehl *et al.*, 2000; Agarwal *et al.*, 2014; Chauhan & Batra, 2018). Hence these three measures of semantic similarity are discussed in further details in the forthcoming paragraphs.

### 2.4.1 Cosine Similarity

Cosine similarity measure is based on classic vector-space model (Inkpen, 2007). Singhal (2001; p.36) explains that “to assign a numeric score to a document for a query, the model measures the similarity between the query vector (since query is also just text and can be

converted into a vector) and the document vector and the angle between two vectors is used as a measure of divergence between the vectors, and cosine of the angle is used as the numeric similarity”. In simpler words, cosine similarity “is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words” (Han *et al.*, 2012).

The cosine similarity between two vectors is computed by their normalized dot product divided by the product of their norm (Orkphol & Yang 2019; p.6). The normalization is usually Euclidean (i.e., the value is normalized to vectors of unit Euclidean length (Sidorov *et al.*, 2014). The cosine similarity measured is in the scale of 0 to 1, and two vectors are said to be similar when the cosine similarity was close to 1, and they were said to be dissimilar when it was close to 0 (Orkphol & Yang 2019).

Given two vectors  $a$  and  $b$ , the cosine similarity measure between them is calculated as shown below (Sidorov *et al.*, 2014; p.492):

the dot product is calculated as

$$a \cdot b = \sum_{i=1}^N a_i b_i$$

the norm is defined as

$$||x|| = \sqrt{x \cdot x}$$

and then the cosine similarity measure is defined as

$$\text{cosine}(a,b) = \frac{a \cdot b}{||a|| \times ||b||}$$

which when given the previous two equations becomes

$$\text{cosine}(a,b) = \frac{\sum_{i=1}^N a_i b_i}{\sqrt{\sum_{i=1}^N a_i^2} \sqrt{\sum_{i=1}^N b_i^2}}$$

Orkphol & Yang 2019 (p.10) cosine similarity only measures the direction of the vector, hence dividing or multiplying a scalar to the resultant word vector affects only the magnitude of the vector and not its direction (except for zero scalars which cancel that word vector out of the sentence vector). For example, in information retrieval and text mining, each term is notionally assigned a different dimension and a document is characterised by a vector where the value of each dimension corresponds to the number of times that term appears in the document, and then cosine similarity then gives a useful measure of how similar two documents are likely to be in terms of their subject matter (Nalawade *et al.*, 2016, p.217).

Cosine similarity is popular because it is very efficient in evaluating, especially for sparse vectors, as only the non-zero dimensions need to be considered (Nalawade *et al.*, 2016, p.217). While listing some of the applications of cosine similarity measure, Thiagarajan and colleagues (2008) state that it has been widely applied in the areas of “content matching scenarios (such as document matching), ontology mapping, document clustering, multimedia search, and as a part of web service matchmaking frameworks”.

One of the disadvantages of cosine similarity is that, “when context information is insufficient, cosine similarity fails to determine the correct word sense” (Orkphol & Yang 2019; p.6). Li & Han (2013) argue that in mathematical perspective, “cosine similarity is perfect”, but if we check it from “the text mining perspective, it may not always be reasonable. Further, Li & Han (2013) believe that overly biased by features with higher values and does not care much about how many features two vectors share, i.e., how many values are zeroes.

Since they believe that proven effectiveness of cosine similarity, Li & Han (2013) decided to derive new metrics by slightly modifying it. They explored distance weighted versions (where distance tends to capture how many features two text segments share) and with extensive experiments on a classical text mining problem, i.e. text classification, they obtained a distance weighted cosine metric (dw-cosine) that performs better than the original cosine metric in most cases.

Since cosine similarity measure lends itself to an intuitive understanding of an issue or question, in an interesting research, Cannon (*et al.* 2018), applied cosine similarity measure to a foreign policy and national security issue of Turkish policy towards the Syrian civil war, by evaluating the information in the reports of the Anadolu Agency (AA), a Turkish state-owned press during the period 2012-2016.

To overcome the limitations of cosine similarity, Sidorov and colleagues (2014) introduced the concept of soft similarity and the soft cosine measure in Vector Space Model (VSM), which into account similarity between features and yielded better results in their research for a question answering task. They use Levenshtein distance (is the number of operations: insertions, deletions, rearrangements needed to convert a string into another string) for similarity of features, because they consider it is a good measure. The soft cosine formula suggested by Sidorov and colleagues (2014; p.494) is mentioned below:

$$soft\_cosine(a,b)=\frac{\sum \sum_{i,j=1}^N a_{ij}b_{ij}}{\sqrt{\sum \sum_{i,j=1}^N a_{ij}^2} \sqrt{\sum \sum_{i,j=1}^N b_{ij}^2}}$$

Sidorov and colleagues (2014; p.494) states that they consider each pair of features as a new feature with a “weight” or “importance” being the similarity of the two features and the advantage of the above formula is its simplicity which can be used with existing machine-learning tools without change, by only recalculating the data vectors.

## 2.4.2 Jaccard Similarity

Jaccard coefficient measures is used for comparing the similarity and diversity of sample sets and it is defined as the size of the intersection divided by the size of the union of the sample sets and the formula used is as follows (Takale & Nandgaonkar, 2010, p.82):

$$J(w_1, w_2)=\frac{|K|_{w_1} \cap K|_{w_2}|}{|K|_{w_1} + K|_{w_2} - K|_{w_1} \cap K|_{w_2}|}$$

Jaccard index was proposed in 1901 as index for the normalization of the binary citation matrix (Jaccard, 1901), which was later modified by Tanimoto (1957) for the non-binary co-citation matrix (Leydesdorff, 2008). Niwattanakul and colleagues (2013) explain that “Jaccard index is a name often used for comparing similarity, dissimilarity, and distance of the data set” and “Jaccard distance is non-similar measurement between data sets, which can

be determined by the inverse of the Jaccard coefficient”. Jaccard coefficient is commonly used in information retrieval as measures of association and it differs from other measures in that it is essentially combinatorial (being based only on sizes of the supports of q, r, and q • r rather than the actual values of the distributions) (Lee, 1999; p.26).

Niwattanakul and colleagues (2013) tested the algorithm to find about Jaccard similarity coefficient by measuring the similarity in the correct grammar syntax and the test of similarity in terms of an error by developing the tests with Prolog programming language and came to the conclusion that Jaccard similarity coefficient is sufficiently suitable to be employed in the word similarity measurement.

Dong & Bhanu (2003, p.4) argue that “one of the advantages of Jaccard coefficient is that it can evaluate a clustering result whose cluster number is not necessarily the true component number”. In the opinion of Bisandu & colleagues (2019, p.3) Jaccard coefficient is “less sensitive to the word swaps, because it considers only whether token exist, not its position” and its evaluation is very efficient and simple. Jaccard coefficient focuses on strong links in segments of the database and it is the best basis for the normalization because this measure does not take the distributions along the respective vectors into account (Leydesdorff, 2008; p.79).

However, the disadvantages of using Jaccard coefficient include typographic errors between tokens are penalized, and the significance of the similarity measure is penalized in case of any error (Bisandu et al. 2019, p.3). Further the Jaccard coefficient does not take into account the shape of the distributions and in the case of the asymmetrical matrix, it “does not exploit the full information contained in the matrix” (Leydesdorff, 2008; p.79, 81).

In their research Mottukuri and colleagues 2016 used extended jaccard coefficient for measuring the similarity between documents. The extended jaccard coefficient for data processing which they believe could be useful in various applications of information retrieval, data mining and web search is mathematically represented by them in the form of below mentioned equation where d1 and d2 are two documents (Mottukuri *et al.* 2016):

$$S_{EJ}(d_1, d_2) = \frac{d_1 \cdot d_2}{d_1 \cdot d_1 + d_2 \cdot d_2 - d_1 \cdot d_2}$$

While comparing the Cosine Similarity and Jaccard Coefficient, Agarwal and colleagues (2014, p.20), came to the following conclusions:

- (1) Time required for cluster generation by using Cosine Similarity measure takes less amount of time as compare to Jaccard Coefficient.
- (2) Similarity cluster generated by Cosine Similarity gives more accurate and relevant result as compare to Jaccard Coefficient.

### 2.4.3 Dice Similarity Coefficient

Proposed first by Lee Raymond Dice in 1945 for measuring of the amount of ecologic association between species (Dice, 1945), Dice similarity coefficient (DSC) evolved as a spatial overlap index and a reproducibility validation metric, where the value of a DSC ranges from 0, indicating no spatial overlap between two sets of binary segmentation results, to 1, indicating complete overlap (Zou *et al.*, 2004). Dice coefficient similarity (DSC) has been a popular metric for evaluating the accuracy of automated or semi-automated

segmentation methods by comparing their results to the ground truth (Andrews & Hamarneh, 2015). It is a similarity measure and for sets  $X$  and  $Y$  of keywords used in information retrieval, the coefficient can be represented in formula as follows (Takale & Nandgaonkar, 2010, p.82):

$$D(w_1, w_2) = 2 \frac{|K|w_1| \cap |K|w_2|}{|K|w_1| + |K|w_2|}$$

Dice coefficient similarity measure is used by researcher because of its simplicity and normalization properties (Montes-y-Gómez et al., 2001). DSC has been widely used in modern medicine for medical image (such as CT Scan, MRI) analysis to achieve high accuracies for various diagnostic purposes encompassing the fields of Biomedical Engineering, Radiology, Oncology, etc. (Winston et al. 2013; Roth et al. 2016).

The foreground is often chosen to be the region of greatest interest, but when the choice of the foreground region is not clear, the DSC suffers from ambiguity as its value differs depending on this choice (Andrews & Hamarneh, 2015). To overcome this problem, in their research experiment Andrews & Hamarneh, (2015) extended the DSC to a continuous function (based on absolute probability differences and the Aitchison distance) which provided a robust and accurate measure of multi-region probabilistic segmentation accuracy.

The Table-1 below compares advantages and disadvantages of various semantic similarity measures (Gupta et al., 2017, p.246)

**Table 1: Comparison of advantages and disadvantages of various semantic similarity measures**

| <b>Method</b>                    | <b>Principle</b>   | <b>Measure</b>         | <b>Feature</b>   | <b>Advantage</b>  | <b>Disadvantage</b>  |
|----------------------------------|--|------------------------|--|---|--|
| <b>Path Based</b>                | <i>Length of the path linking different word senses</i>    | <i>Shortest Path</i>   | <i>Number of edges between the concepts</i>                                    | <i>Simple measure</i>   | <i>Different pairs of equal length and shortest path will have same similarity</i>                     |
|                                  |  | <i>Wu &amp; Palmer</i> | <i>Path length augmented by subsume path to root</i>                           | <i>Simple measure</i>   | <i>Different pairs having lowest common subsume and equal length of path will have same similarity</i> |
|                                  |  | <i>L &amp; C</i>       | <i>Number of edges between the concepts</i>                                    | <i>Simple measure</i>   | <i>Different pairs of equal length and shortest path will have same similarity</i>                     |
| <b>Information Content Board</b> | <i>The concepts sharing common information are similar</i> | <i>Resnik</i>          | <i>Information content of the lowest common subsumer</i>                       | <i>Simple measure</i>   | <i>Different pairs having lowest common subsume will have same similarity</i>                          |
|                                  |  | <i>Lin</i>             | <i>Information content of the lowest common subsumer and compared concepts</i> | <i>Considers the information content of compared concepts</i> | <i>Different pair having the same summation of information content will have same similarity</i>       |

|                      |  |                |   |  |                                |
|----------------------|--|----------------|---|--|--------------------------------|
| <b>Feature Based</b> | <i>The concepts having common features are similar</i> | <i>Tversky</i> | <i>Compares features of the concept</i> | <i>Considers features while computing similarity</i> | <i>Computationally complex</i> |
|----------------------|--|----------------|---|--|--------------------------------|

Harispe et al. (2015, p. 96) suggest “accuracy, precision and robustness, computational complexity (e.g., algorithmic complexity), mathematical properties, semantics, characterisation regarding technical details as some of the criteria that can be used for evaluating semantic measures”. Measures of semantic similarity and relatedness can improve the performance of information retrieval (IR) and document retrieval (DR) application systems (Pedersen, et al. 2007). Various usage of semantic similarity measures in natural language processing applications include, automatic creation of thesauri, automatic indexing, text annotation and summarization, text classification, word sense disambiguation, information extraction and retrieval, lexical selection, automatic correction of word errors in text, discovering word senses directly from text and language modeling by grouping similar words into classes (Inkpen, 2007, p.12). In an interesting research on semantic similarity, Inkpen (2007, p.18) developed an Intelligent Thesaurus (a writing aid tool) which presents to the writer a set of synonym in the order of priority, and helps the user to choose the synonym which is most appropriate to the context.

Gabrilovich & Markovitch (2007) proposed an Explicit Semantic Analysis (ESA), for fine-grained semantic representation of unrestricted natural language texts (a high-dimensional space of natural concepts derived from Wikipedia), which circumvents the interpretation problems present in the Latent Semantic Models. Similarly, Benedetti and colleagues (2018, p.136) has proposed a unique semantic technique called Context Semantic Analysis (CSA) for “estimating inter-document similarity, leveraging the information contained in a knowledge base and one of the main features of CSA with respect other knowledge-based approaches is its applicability over any RDF knowledge base, so that all datasets belonging to the LOD cloud (more than one thousand) can be used”.

## 2.5 Sentence Semantic Similarity

Intelligent information processing and making sense of the large volume of digital literature has become an urgent need with the rapid growth of available information on the internet. While processing textual information, in the hierarchy of word-sentence-paragraph-document, sentences as the intermediate blocks are crucial to understanding the semantics of the text (Chen *et al.* 2018). By definition, “Semantic Measures are mathematical tools used to estimate quantitatively or qualitatively the strength of the semantic relationship between units of language, concepts or instances, through a numerical or symbolic description obtained according to the comparison of information formally or implicitly supporting their meaning or describing their nature” (Harispe *et al.*, p.12). Sentence Semantic Similarity is a measure of conceptual distance between sentences, based on the correspondence of their meanings (Pawar & Mago, 2017). Stated in simple terms sentence semantic similarity is a procedure to determine how related or unrelated two sentences are (Sharma & Srivastava, 2017). Sentence semantic similarity is an approach vital to our language understanding (Ru *et al.*, 2013). Hence, computing the semantic similarity between sentences has become an important component in many natural language processing tasks (Soğancıoğlu, *et al.*, 2017). It provides the foundation for various applications related to machine translation, text summarization, text categorization, question answering, short answer grading, semantic search,

conversational systems (Cer *et al.*, 2018) biomedical informatics and geoinformation (Majumder *et al.* 2016). Ru & colleagues (2013) argue that sentence semantic similarity provides a practical alternative to true understanding of languages, since it requires world knowledge (as in humans), a yet to-be-solved problem in Artificial Intelligence.

Traditionally the methods of measuring Sentence Semantic Similarity was based on lexical semantics, surface form matching and basic syntactic similarity and subsequently alignment based method and deep learning methods were developed (Cer *et al.*, 2018). Since sentences are made of a set of words, ontology-based word-level similarity measures can be used to compute semantic similarity scores between sentences (Soğancıoğlu, *et al.* 2017, p.52).

Sentence Semantic Similarity software captures degrees of semantic equivalence and the objective is to create a unified framework for extracting and measuring semantic similarity, thus replicating human language understanding (Rychalska *et al.*, 2017). According to the concrete knowledge sources exploited and the way, in which they are used, different families of methods can be classified as follows (Martinez-Gil 2014, p.936; Jiang *et al.* 2017, p.249):

- (1) Edge counting measures: which consist of taking into account the length of the path linking the concepts (or terms) and the position of the concepts (or terms) in a given dictionary (or taxonomy, ontology);
- (2) Feature based measures: which consist of measuring the similarity between concepts (or terms) as a function of their properties or based on their relationships to other similar concepts (or terms);
- (3) Information content measures: which consist of measuring the difference of the information content of the two concepts (or terms) as a function of their probability of occurrence in a text corpus (or an ontology)
- (4) Hybrid measures: which consist of combining all of the above.

Further, Martinez-Gil (2014, p.937) categorizes the methods for measuring semantic similarity using web search engines as follows:

1. Co-occurrence methods: which consist of measuring the probability of co-occurrence of the terms on the Web.
2. Frequent patterns finding methods: which consist of finding similarity patterns in the content indexed by the web search engine.
3. Text snippet comparison methods: which consist of determining the similarity of the text snippets from the search engines for each term pair.
4. Trend analysis methods: which consist of comparing the time series representing the historical searches for the terms.

Sentence similarity was measured by Mihalcea and colleagues (2006, p.776) as a function of the semantic similarity of the component words, by combining metrics of word-to-word similarity and word specificity into a formula that is a potentially good indicator of the semantic similarity of the two input texts. They came to the conclusion that this measure of text semantic similarity outperforms the simpler vector-based similarity approach, as evaluated on a paraphrase recognition task.

Since words convey meaning in a sentence, they are tagged with appropriate senses initially and then sentence similarity is calculated based on the number of shared senses (Xu & Lu, 2013). In this method, to capture the meaning between sentences, Xu & Lu (2013) disambiguate word senses using contexts and then determine sentence similarity by counting the senses they shared.



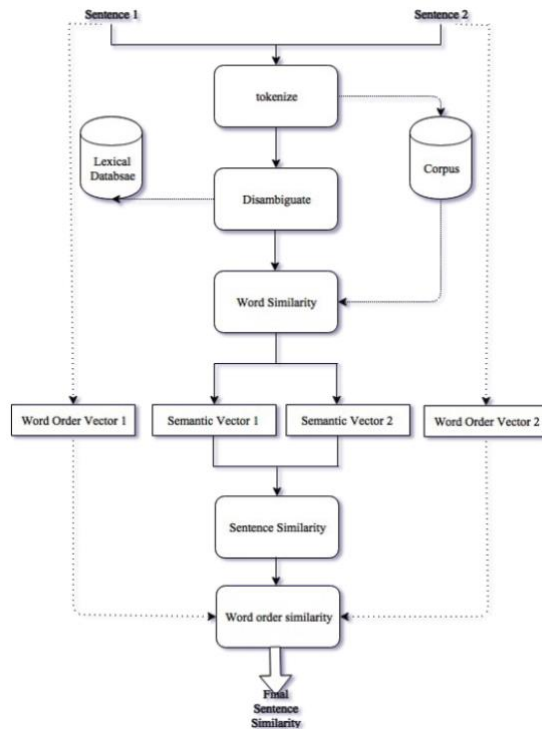
Some of the state-of-the-art publicly available tools for generic domain sentence semantic similarity computation are described below (Sogancioglu, *et al.*, 2017, p. 50):

1. ADW (Align, Disambiguate and Walk): ADW an open source, unified approach for computing semantic similarity was proposed by Pilehvar & colleagues (2013; p.1349) that leverages a common probabilistic representation at the sense level for all types of linguistic data. ADW is a knowledge-based system that uses the Topic-sensitive PageRank algorithm over a graph generated using WordNet to model the semantic similarity between words, phrases, sentences, and documents. Pilehvar and Navigli (2015) describe the three main advantages of ADW as: (1) it is applicable to all types of linguistic items, from word senses to texts; (2) it is all-in-one, i.e., it does not need any additional resource, training or tuning; and (3) it has proven to be highly reliable at different lexical levels and multiple evaluation benchmarks.
2. SEMILAR (SEMantic simILARity): SEMILAR is a toolkit developed by Rus & colleagues (2013), implements a number of algorithms for assessing the semantic similarity between two texts is available as a Java library and as a Java standalone application offering GUI-based access to the implemented semantic similarity methods. Rus & colleagues (2013, p.164) claim that SEMILAR is a “one-stop-shop for investigating, annotating, and authoring methods for the semantic similarity of texts of any level of granularity”.

Some of the limitations associated with the current methods measuring sentence semantic similarity are as follows (Lee *et al.*, 2014, p2):

- (1) The conventional methods assume that a document has hundreds or thousands of dimensions, transferring the short texts/sentences into a very high dimensional space and extremely sparse vectors may lead to a less accurate calculation result.
- (2) Algorithms based on shared terms are suitable to be applied to the retrieval of medium and longer texts that contain more information. In contrast, information of shared terms in short texts or sentences is rare and even inaccessible. This may cause the system to generate a very low score on semantic similarity, and this result cannot be adjusted by a general smoothing function.
- (3) Stop words are usually not taken into consideration in the indexing of normal IR systems. Stop words do not have much meaning when calculating the similarity between longer texts. However, they are unavoidable parts with regard to the similarity between sentences, for that they deliver information concerning the structure of sentences, which has a certain degree of impact on explaining the meanings of sentences.
- (4) Similar sentences may be composed of synonyms; abundant shared terms are not necessary. Current studies evaluate similarity according to the co-occurring terms in the texts and ignore syntactic information.

To overcome the limitations of various methods currently in vogue, Pawar & Mago (2018, p.2) proposes an algorithm which works by “disambiguating the words in sentences and forming semantic vectors dynamically for the compared sentences and words”. In their methodology, text is considered as a sequence of words and the words in sentences are dealt with separately “according to their semantic and syntactic structure and the information content of the word is related to the frequency of the meaning of the word in a lexical database or a corpus” (Pawar & Mago (2018, p.2).

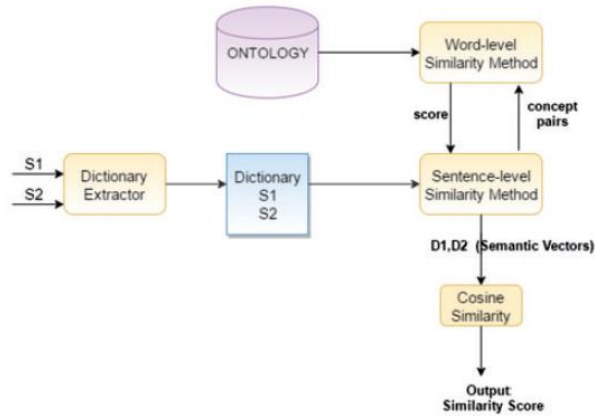


**Figure-4: Process to calculate the similarity between two sentences (Pawar & Mago, 2018, p.2)**

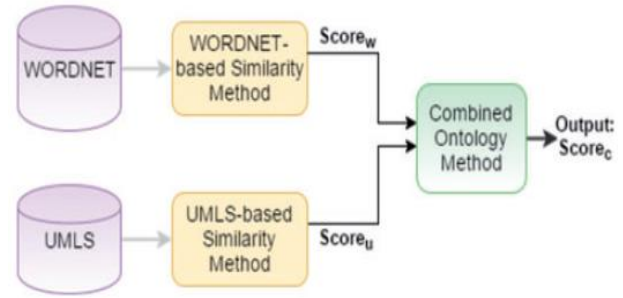
The method shown in Figure-4, works in four steps described below (Pawar & Mago, 2018):

1. Word similarity: calculated using established edge-based approach, identifying words for comparison, associating word for making sense, calculating the shortest path distance between synsets and establishing the hierarchical distribution of words.
2. Information content of the word analysed: Finding the meaning of the word related to the specific context.
3. Sentence semantic similarity: calculate the semantic similarity measure between sentences using the semantic value vectors.
4. Word order similarity estimated: treating treat sentences relatively to keep the size of vector minimum. (The word order similarity actually matters when two sentences contain same words in different order but if the sentences contain different words, the word order similarity should be an optional construct).

Soğancıoğlu & colleagues (2017, p.i53-54) introduced a sentence-level ontology-based methods namely WordNet-based Similarity Method (WBSM), UMLS-based Similarity Method (UBSM) and combined ontology method (COM) as described below, specifically for the biomedical domain.



**Figure-5: Sentence-level similarity module method**



**Figure-6: Sentence-level combined ontology**

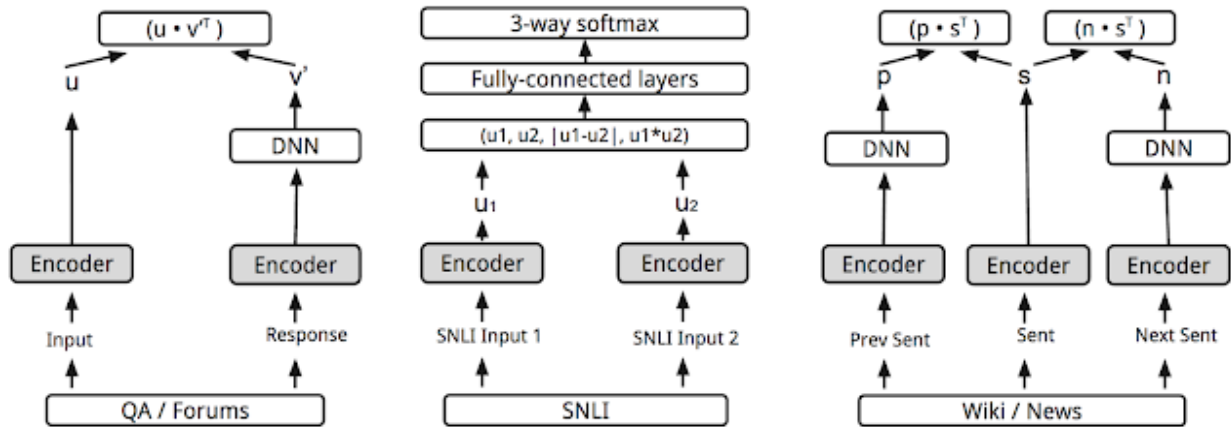
[NOTE: S1 and S2 are two sentences, dictionary D is constructed, which consists of the union of the unique words from the two sentences. Dis used to build the semantic vectors D1 and D2 for S1 and S2, respectively, which have the same dimension as the dictionary and finally, the cosine similarity between D1 and D2 gives the semantic similarity score between the two sentences S1 and S2.]

Calculation of word-level similarities and adapting word-level similarities to obtain sentence-level score (sentence-level similarity method) are two main tasks in the general flow (Sogancioglu, *et al.*, 2017, p.i54). They developed a sentence-level similarity method, which is an algorithm to adapt word-level similarities to sentence-level.

Conneau & colleagues (2017, p.670) from Facebook Research AI, while studying the task of learning universal representations of sentences, they compare sentence embeddings trained on various supervised tasks, and show that sentence embeddings generated from models trained on a natural language inference (NLI) task reach the best results in terms of transfer accuracy and they came to the conclusion that universal sentence representations trained using the supervised data of the Stanford Natural Language Inference datasets can consistently outperform unsupervised methods like SkipThought vectors.

### 2.5.1 Google Universal Sentence Encoder

Cer & colleagues (2018) from Google Research presented two models of Universal Sentence Encoder (USE) for encoding sentences into embedding vectors, one using the transformer architecture, while the other is formulated as a deep averaging network and both models are implemented in TensorFlow. Further Yang & Tar (2018) explain that these models “extends the multitask training described above by adding more tasks, jointly training them with a skip-thought-like model that predicts sentences surrounding a given selection of text”.



**Figure-7: Multi-task training as described in “Universal Sentence Encoder”. A variety of tasks and task structures are joined by shared encoder layers/parameters (grey boxes) [Yang & Tar, 2018].**

In these two models, input is English strings and output is a fixed dimensional embedding representation of the string. Computation of sentence level semantic similarity scores, in these models, using the sentence embeddings achieve excellent performance on the semantic textual similarity (STS) Benchmark (Cer *et al.*, 2018). The main aim of USE is “to provide a single encoder that can support as wide a variety of applications as possible, including paraphrase detection, relatedness, clustering and custom text classification” (Yang & Tar, 2018).

In addition to the above USE models, in May 2018, Yang & Tar (2018) from Google AI, introduced two *new* models on TensorFlow Hub as described below:

- 1) Universal Sentence Encoder-Large: The Large model is trained with the Transformer encoder and it targets scenarios requiring high precision semantic representations and the best model performance at the cost of speed & size.
- 2) Universal Sentence Encoder-Lite: The Lite model is trained on a Sentence Piece vocabulary instead of words in order to significantly reduce the vocabulary size, and it targets scenarios where resources like memory and CPU are limited, such as on-device or browser based implementations.

The above two models that return a semantic encoding for variable-length text inputs and can be used for semantic similarity measurement, relatedness, classification, or clustering of natural language text” (Yang & Tar, 2018).

### 2.5.2 Microsoft Academic Similarity

Microsoft Academic (MA) is a free academic search engine and a citation index (Thelwall, 2018, p.1). MA was officially launched in July 2017 that claims to include records for over 170 million scholarly publications from publisher websites, authors' personal homepages and documents indexed by the Bing search engine (Kousha *et al.*, 2018, p.289). It is a promising new data source for evaluative bibliometrics due to its size and functionality (Scheidsteger *et al.* 2018). The earliest version of Microsoft Academic called Windows Live Academic was launched in 2006, which had a database of around eight million articles from various scientific disciplines e.g. computer science, electrical engineering, and physics (Carlson, 2006). It was renamed as Live Search Academic in 2008 and was scrapped in 2010 when

Microsoft found that it did not have sufficient development support (Fagan 2017). Finally, it was converted into Microsoft Academic Search (MAS) after a complete redesign of the service carried out by its affiliate, the Microsoft Asia Research Group in China (Orduña Malea *et al.*, 2014). Early reviews of the 2011 edition of Microsoft Academic Search got promising early reviews in 2011, but it clearly lacked the quantity of data searched by Google Scholar (Fagan 2017). MAS is a scientific web database which gathers bibliographic information from the principal scientific editorials (Elsevier, Springer) and bibliographic services (CrossRef) (Ortega and Aguillo, 2014). The core of Microsoft Academic Service (MAS) is a heterogeneous entity graph comprised of six types of entities that model the scholarly activities: field of study, author, institution, paper, venue, and event (Sinha *et al.* 2015).

Sinha and colleagues (2015) at the Microsoft Research, Redmond (USA) proposed a modified version of Microsoft Academic Search (MAS) to include data mining results from the Web index and an in-house knowledge base from Bing (search engine), in addition to obtaining six entities (mentioned above) from the publisher feeds. In the new version as a result of the Bing integration (Sinha *et al.* 2018) the new MAS graph sees significant increase in size (for instance, the number of papers indexed by MAS grew from low tens of millions to 83 million while maintaining an above 95% accuracy based on test data). They explain that this growth happened due to fresh information streaming in automatically following their discoveries by the search engine. Further advantage of this new version is that the rich entity relations included in the knowledge base provide additional signals to disambiguate and enrich the entities within and beyond the academic domain.

While evaluating the suitability of MAS and Google Scholar Citations (GSC) for bibliometric researches, Ortega and Aguillo (2014; p,1154) conclude that MAS is better recommended for disciplinary studies than for analyses at institutional and individual levels, whereas GSC is a good tool for individual assessment because it counts on a wider variety of documents and citations.

Microsoft Academic Knowledge Applications Programming Interface (API) consists of four related points<sup>9</sup>:

1. Interpret: Interprets a natural language user query string. Returns annotated interpretations to enable rich search-box auto-completion experiences that anticipate what the user is typing.
2. Evaluate: Evaluates a query expression and returns Academic Knowledge entity results.
3. Calchistogram: Calculates a histogram of the distribution of attribute values for the academic entities returned by a query expression, such as the distribution of citations by year for a given author.

Used together, these API methods allow the user to create a rich semantic search experience. Given a user query string, the interpret method provides an annotated version of the query and a structured query expression, while optionally completing the user's query based on the semantics of the underlying academic data. For example, if a user types the string *latents*, the interpret method can provide a set of ranked interpretations, suggesting that the user might be searching for the field of study *latent semantic analysis*, the paper *latent structure analysis*, or other entity expressions starting with *latents*. User can use this information to quickly reach the desired search results. To retrieve a set of matching paper entities from the academic knowledge base, the evaluate method can be used and to calculate the distribution

<sup>9</sup> <https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/home>

of attribute values for a set of paper entities which can be used to further filter the search results, the calchistogram method can be used<sup>10</sup>.

In further improvements, Microsoft Academic increased power of semantic search by adding more fields of study<sup>11</sup>. As a consequence, Microsoft Academic has now become a valuable source of citation data for both papers and books and it is a good database for conducting citation context studies because it has made it possible to download citation contexts that are already segmented (Tahamtan & Bornmann, 2019). Further, Bornmann and colleagues (2019) argue that, availability of large amounts of citation context data in bibliometric databases such as Microsoft Academic (MA) makes it possible to investigate concept symbols in large datasets. Hug *et al.* (2016, p.8) while examining Microsoft's Academic Applications Programming Interface (an interface to access MA data) vis-à-vis Google Scholar, in September 2016, found that it offers rich, structured metadata with the exception of document type and Microsoft Academic has "grown massively from 83 million publication records in 2015 to 140 million in 2016".

But some researchers such as Ward *et al.* (2015, p.190) had argued that although Microsoft Academic Search offers "a pretty profile page, pre-fabricated with basic metrics and charts", but "the author gains at the setup is easily lost while trying to consolidate data, correct items wrongfully attributed to him or her, and add missing items". Fagan (2017) expresses the concern that Microsoft Academic appears to have less coverage of books and grey literature compared with Google Scholar. Looking critically at the features of Microsoft Academic, Hug *et al.* (2016, p.9) identified the four major limitations of the available metadata, which are mentioned below:

- a) First, MA does not provide the document type of a publication.
- b) Second, the "fields of study" are dynamic, too specific and field hierarchies are incoherent.
- c) Third, some publications are assigned to incorrect years.
- d) Fourth, the metadata of some publications did not include all authors.

In an attempt to build a bibliometric view, Ortega (2014) analysed the Microsoft Academic and came to the conclusion that it could be a reliable tool for collaboration studies if the limitations are previous addressed, concretely the cleansing of duplicated profiles.

The creators of Microsoft Academic admit that the publication metadata is neither complete nor accurate and they state that some of the common problems include: most papers about a topic like "artificial intelligence" do not actually mention these words explicitly in the paper (incomplete); a large number of raw keywords from various data sources are noisy and irrelevant to the paper (inaccuracy, e.g. some websites assigned same sets of keywords to all papers published on it); same words refer to different concepts in different disciplines (ambiguity, e.g., "entropy")<sup>12</sup>.

Microsoft Academic team applied several state-of-the-art natural language processing techniques to tackle these challenges. For example, we extended convolutional neural

<sup>10</sup> <https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/home>

<sup>11</sup> <https://www.microsoft.com/en-us/research/project/academic/articles/microsoft-academic-increases-power-semantic-search-adding-fields-study/>

<sup>12</sup> <https://www.microsoft.com/en-us/research/project/academic/articles/microsoft-academic-increases-power-semantic-search-adding-fields-study/>



networks for short text classification and made it highly scalable for our 140M English papers, such that high-level disciplines such as: computer science, mathematics, artificial intelligence, etc., would be properly tagged. They also pre-trained word embedding vectors with text from more than 80M abstracts, used together with bag-of-words for text similarity calculation, this helped to eliminate noisy tagging effectively. Application of these techniques resulted in high accuracy to their observations<sup>13</sup>.

In its latest version Microsoft Academic has become a powerful tool which makes it possible to discover knowledge mainly because of the three aspects that contribute to the power of Microsoft Academic's semantic search<sup>14</sup>:

1. Author entity disambiguation, addressed in a previous post;
2. The recent increase in number of fields of study in our graph, and
3. The accuracy of tagging fields of study onto papers.

The openness and the abilities of Microsoft Academic for supporting scholarly social networking is a useful feature (Fagan, 2017). In a comparative study of Microsoft Academic Service, Web of Science, Scopus, and Google Scholar, the author comes to the conclusion that Microsoft Academic Service “might well turn out to combine the advantages of broader coverage, as displayed by Google Scholar, with the advantage of a more structured approach to data presentation, typical of Scopus and the Web of Science” (Harzing, 2016). Microsoft Academic has advantage over other search engines (such as Google Scholar) in that, it allows automated searching through its Applications Programming Interface (API), and provides better performance when it comes to showing the citation context of papers and other information (Tahamtan & Bornmann, 2019).

## 2.6 Text Semantics

Text mining techniques traditionally, have relied on both a bag-of-words model (syntax model) and application of data mining techniques, whereby, only the lexical component of the texts are considered (Sinoara et al. 2017). This approach is reasonable for tasks requiring routine information extraction such as search and categorization (Sebastiani, 2002). Sinoara et al. (2017) suggest that there has been increasing interest of text mining researchers in text semantics to get a more complete analysis of text collections and better text mining results. Progress of the computing capacity, (which is continuously reducing the processing time) and the recent NLP developments now allow processing of raw texts at a much deeper level.

According to Jay Lemke (1995; p.90) text semantics is a distinct model of semantics in “which larger discourse wholes contextualize the meanings of grammatical structures (e.g. clause like units) and words”. In the present context, text semantics, as Sinoara & colleagues (2017) explains, can be considered in the three main steps of text mining process:

- 1) Pre-processing: data representation can be based on semantic aspect of text collection
- 2) Pattern extraction: semantic information can be used to guide the model generation or to refine it
- 3) Post-processing: extracted patterns can be evaluated based on semantic aspects.

<sup>13</sup> <https://www.microsoft.com/en-us/research/project/academic/articles/microsoft-academic-increases-power-semantic-search-adding-fields-study/>

<sup>14</sup> <https://www.microsoft.com/en-us/research/project/academic/articles/microsoft-academic-increases-power-semantic-search-adding-fields-study/>

Text semantics characterizes “discursive formations at an appropriate level of abstraction from specific texts, to describe how discursive formations are instanced and combined in texts” and it concerns itself “primarily with meaning relations within and between texts that are not made at the rank of the clause or below” (Lemke, 1995; p.90).

Collection of large documents can be organised into smaller meaningful and manageable groups using a useful technique called text clustering, but traditional text clustering algorithms relying on the BOW (Bag of Words) approach, has an obvious disadvantage that it ignores the semantic relationship among words so that it cannot accurately represent the meaning of documents Wei et al. (2015). Further Cambria & White (2014, p. 48) believes that in information extraction research, “most of the existing approaches are still based on the syntactic representation of text, a method that relies mainly on word co-occurrence frequencies” and “such algorithms are limited by the fact that they can process only the information that they can ‘see’”. Shifting from syntactic information (e.g. part-of-speech tagging, chunking, and parsing) to semantic information (e.g. word-sense disambiguation, semantic role labeling, named entity extraction, and anaphora resolution) would enable us to construct software systems for real-world solutions (Collobert *et al.* 2011).

In their book on ‘Sentic Computing’, Cambria & Hussain (2012, p.21) put forward the argument that instead of simply counting word co-occurrence frequencies in text adoption of the bag-of-concepts model would enable working at concept level entails preserving the meaning carried by multi-word expressions, with the added advantage that this model also “helps to overcome problems such as word-sense disambiguation and semantic role labelling”.

Although the lexical and compositional semantics are equally necessary with regard to our understanding of languages<sup>15</sup>, the current literature indicates that the NLP research is “gradually shifting from lexical semantics to compositional semantics” and ultimately pragmatics will enable NLP to be “more adaptive and, hence open-domain, context-aware, and intent-driven to evolve into natural language understanding evolve into Natural Language Understanding” (Cambria & White, 2014, p.52& p.56). Perrián-Pascual and Arcas-Túnez (2007) argue that “when meaning postulates become more complex cognitively, there is no way to state co-reference between internal conceptual units just via semantic relations”. Therefore, in future, the NLP research has to move towards pragmatic curve characterized by “bag-of-narratives model, whereby each piece of text will be represented by mini-stories or interconnected episodes”, leading to a more detailed level of text comprehension and sensible computation, thereby “tackling NLP issues such as co-reference resolution and textual entailment”. (Cambria & Hussain, 2012, p.20).

Using their system (named IntelliZap), Finkelstein & colleagues (2002) conducted a research keyword-based search engines following context-driven information retrieval process which involves semantic keyword extraction. The four main components of IntelliZap: context capturing (performed by client-side software), extracting keywords from the captured text and context, high-level classification of the query to a small set of predefined domains, re-ranking the results obtained from different search engines. Results of their experiment suggested that using context to guide search, effectively offers even inexperienced users an advanced search tool on the web.

<sup>15</sup> <https://brotherfish.me/portfolio/lexical-semantics/>



Adapting a number of measures of similarity and relatedness to the biomedical domain, Pedersen & colleagues (2007) found that there is a role both for more flexible measures of relatedness based on information derived from corpora, as well as for measures that rely on existing ontological structures. In another research of semantic similarity and relatedness in biomedical domain, Henry & colleagues (2019, p.1) use “co-occurrence statistics between Unified Medical Language System (UMLS) concepts to account for lexical variation at the synonymous level, and introduce a process of concept expansion that exploits hierarchical information from the UMLS to account for lexical variation at the hyponymous level”. Patient’s opinions about medicines and doctors written in Spanish are analyzed by Jiménez-Zafra & colleagues (2019) applying supervised learning and lexicon-based sentiment analysis approaches. In the opinion expressed about medicines, the researchers found greater lexical diversity (making the task of sentiment analysis difficult) but the patient’s opinion about the doctors were more specific (good or bad medical practice).

In their information extraction research, Sil et al. (2010, p.108), used their PREPOST system (which combines traditional text mining techniques with open-domain semantic role labeling to mine knowledge about action semantics) and demonstrated that PREPOST can identify the preconditions of previously unseen actions using automatically downloaded Web documents with a precision of over 80% at 100% recall (80% precision at 77% recall for effects”).

## 2.7 Stemming and Lemmatization

Stemming and Lemmatization are normalization techniques that are very useful for the purpose of finding a connection between related words or word forms and such a normalization is important in various NLP applications, like text classification and information extraction, because it brings out actual grammatical or semantic relations which are otherwise not accessible by the software (Ingason *et al.*, 2008). These two are important natural language processing techniques widely used in Information Retrieval (IR) for query processing and in Machine Translation (MT) for reducing the data sparseness (Gupta *et al.* 2012). Given below, first *stemming* is discussed, followed by *lemmatization*.

### 2.7.1 Stemming

Stemming usually refers to a crude heuristic process that cuts the ends of words in the hope of achieving this goal (correctly most of the time), and often includes the removal of derivational affixes<sup>16</sup>. To improve retrieval effectiveness and to reduce the size of indexing files, stemming is used to achieve compression factor of over 50 percent (Frakes,1992).

J. B. Lovins was the first to present a stemming algorithm for information-retrieval applications and introduced the idea of stemming based on a dictionary of common suffixes, such as \*SES, \*ING or \*ATION (Willett, 2006). In Lovin’s two-phase system, in the first phase, the stemming algorithm proper, retrieves the stem of a word by removing its longest possible ending which matches one on a list stored in the computer and in the second phase handles “spelling exceptions”, mostly instances in which the same stem varies slightly in spelling according to what suffixes originally followed it (Lovins, 1968).

According to Lovins (1968, p.1), a “stemming algorithm is a computational procedure which reduces all words with the same root (or, if pre-fixes are left untouched, the same *stem*) to a

<sup>16</sup> <https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>

common form, usually by stripping each word of its derivational and inflectional suffixes”. When a word is presented for stemming, in a dictionary-based stemming algorithm, the right-hand end of the word is checked for the presence of any of the suffixes in the dictionary and if a suffix is found to be present, it is removed, subject to a range of context-sensitive rules that forbid such removal (Willett, 2006).

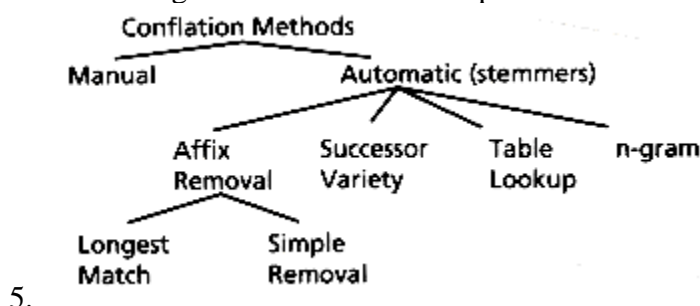
*Iteration and Longest-match* are the two main principles are used in the construction of a stemming algorithm Lovins (1968):

- (1) Iteration: Iterative stemming algorithm is a recursive procedure based on the fact that suffixes are attached to stems in a certain order, that is, there exist *order-classes* of suffixes and it removes strings in each order-class one at a time, starting at the end of a word and working toward its beginning.
- (2) Longest-match: The *longest-match* principle states that within any given class of endings, if more than one ending provides a match, the one which is longest should be removed. This principle is implemented by scanning the endings in any class in order of decreasing length.

Lovins (1968) adds a caution, that an algorithm based solely on one of these methods often has drawbacks which can be offset by employing some combination of the two principles. Balakrishnan and Llyod-Yemoh (2014, p.175) argues that the “Lovin’s stemmer is a single pass, context-sensitive algorithm which only removes one suffix from a word by utilizing a list of 250 suffixes and removing the longest suffix that it finds attached to the given word and it ensures that when a word has been stemmed, it is at least three characters long”. Lovin’s stemmer deals with both information retrieval and computational linguistics problems.

The taxonomy for stemming algorithms is explained in Figure 1, below and the four automatic approaches are (Frakes,1992):

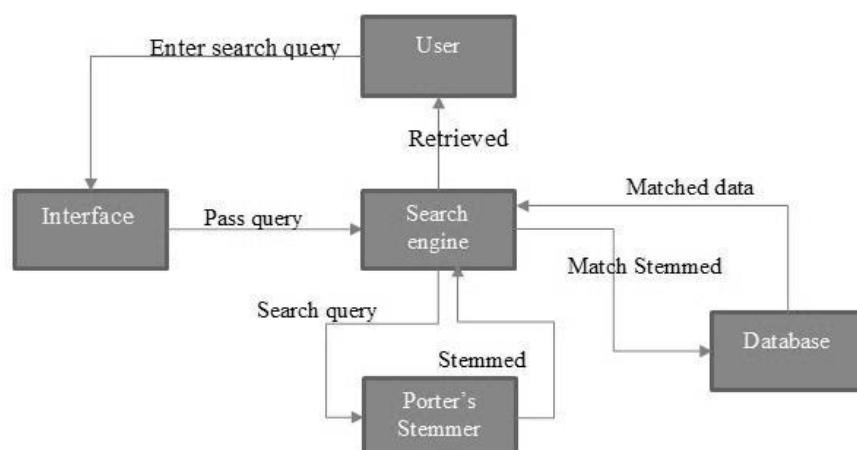
1. Affix removal algorithms remove suffixes and/or prefixes from terms leaving a *stem* and sometimes transform the resultant stem. This being the most common approach, the name *stemming* derives from this method.
2. Successor variety stemmers use the frequencies of letter sequences in a body of text as the basis of stemming.
3. The n-gram method conflates terms based on the number of diagrams or n-grams they share. Terms and their corresponding stems can also be stored in a table.
4. Stemming is then done via lookups in the table.



**Figure 8: Conflation methods (Source: Frakes,1992)**

Porter’s stemmer is one of the most commonly used stemmer for information retrieval (Balakrishnan and Llyod-Yemoh, 2014). The Porter algorithm consists of a set of condition/action rules and the conditions fall into three classes: conditions on the stem,

conditions on the suffix, and conditions on the rules (Frakes, 1992). How the Porter's stemmer works is explained in Figure 9 below:



**Figure 9: Data flow diagram for stemming (Source: Balakrishnan and Llyod-Yemoh, 2014)**

The goal of both stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form<sup>17</sup>. For instances Porter (1980) uses the following example to illustrate stemming:

*connect, connected, connection, connecting* ⇒ *connect*

By stripping the root of its derivational and inflectional affixes, a stemming algorithm reduces all words with the same root to a single form, the *stem* (Willett, 2006). The main merits of the stemming program are: it is small, fast and reasonably simple (Porter, 1980).

Gupta *et al.* (2012) argue that stemming is normally exposed to two problems as described below:

- (1) Over-stemming: it occurs when words that are not morphological variants are conflated (i.e. in case of conflation of semantically distant words).
- (2) Under-stemming: it occurs when words that are morphological variants are not conflated (i.e. when two semantically exact words which may be differently inflected should be stemmed).

In various fields of computational linguistics and information retrieval, researchers find this a desirable step, for reasons such as the root of a word may be of less immediate interest than its suffixes, which can be used as clues to grammatical structure, in automated morphological analysis (Lovins, 1968).

## 2.7.2 Lemmatization

Malaviya & colleagues (2019, p.1) define *lemmatization* as “a core NLP task that involves a string-to-string transduction from an inflected word form to its citation form, known as the *lemma*”. In other words, *lemmatization* is the process to determine the original form of the

<sup>17</sup> <https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>

dictionary word which is known as the *lemma*<sup>18</sup> (e.g. run) by the morphological analysis of the inflectional variations of a word (e.g. ran, runs, running). Balakrishnan and Llyod-Yemoh (2014) explains that lemmatization uses vocabulary and morphological analysis of word and removes inflectional endings, thereby returning words to their dictionary form and also helps to match synonyms by the use of a thesaurus.

Manjavacas and colleagues (2019) states that lemmatization of standard languages is concerned with (i) abstracting over morphological differences and (ii) resolving token-lemma ambiguities of inflected words in order to map them to a dictionary headword. For languages with rich inflectional morphology, lemmatization is one of the basic and indispensable steps in natural language processing (Chrupała, 2006). Especially for languages with higher surface variation, lemmatization plays an important role as a preprocessing step for downstream tasks such as topic modeling, and information retrieval. (Manjavacas *et al.* 2019). Lemmatization is crucial for many NLP tasks and machine translation (Muller *et al.*, 2015). In information retrieval process and other NLP applications for languages with rich morphology, lemmatization is used as a preprocessing step and has been shown to outperform stemming for some tasks (Barteld *et al.*, 2016). Among the several advantages over alternative lexicon or rule based methods, lemmatization requires much less effort on the part of human experts and is to a large extent language-independent (Chrupała, 2006). Whenever it is required to map words to lexical resources and establish the relation between inflected forms, particularly critical for morphologically rich languages, lemmatization is very useful (Muller *et al.*, 2015).

In modern data-driven approaches, lemmatization is treated as a classification task where classes are represented by binary edit-trees induced from the training data (Manjavacas *et al.* 2019). One of the main purposes of data-driven lemmatization is to handle unseen words at test time, yet languages with differing morphological productivity will have very different proportions of unseen words (Bergmanis & Goldwater, 2018). Normally data driven lemmatization are used as means to tackle some of the issues associated with morphologically-rich languages such as: lexical data sparseness that originates from rich inflections and the small size of syntactically annotated data available for such languages (Seddah, 2012). Explaining the data-driven approaches to lemmatization process, Manjavacas and colleagues (2019) states that given a token lemma pair, its binary edit-tree is induced by computing the prefix and suffix around the longest common subsequence, and recursively building a tree until no common character can be found and such edit-trees manage to capture a large proportion of the morphological regularity, especially for languages that rely on suffixation for morphological inflection. Ingason *et al.* (2008), adopted a Hierarchy of Linguistic Identities (HOLI) approach for developing an Icelandic NLP tool called *Lemmald* which uses an algorithm for lemmatizing morphologically rich languages, combining data-driven machine learning methods and linguistic knowledge, which achieves good performance by relying on IceTagger for tagging and The Icelandic Frequency Dictionary corpus for training.

Bergmanis & Goldwater (2018, p.1391) argue that the two main challenges faced by data-driven lemmatizers are: first, to generalize beyond the training data in order to lemmatize unknown words; and second, to disambiguate ambiguous wordforms from their sentence context. Erjavec and Dzeroski (2004) suggested a two-stage architecture, first sentences are assigned morpho-tag sequences by a POS-tagger, and then an Inductive Logic Programming

<sup>18</sup> <https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>

system assigns lemmas to unknown wordform-tag pairs to solve the problem of lemmatizing unknown words (Chrupała et al. 2008).

To illustrate how lemmatization is context sensitive, Malaviya et al. (2019) use the simple example of multiple forms of English verb (e.g. *talk* may also appear as *talks*, *talked* or *talking*, depending on the context). Chakrabarty *et al.* (2017) explains that context sensitive lemmatization is used to handle diverse text processing problems (e.g. sense disambiguation, parsing, translation), for context sensitive languages (i.e. where same inflected word form may come from different sources and can only be disambiguated by considering its neighbouring information). Highlighting the benefits of context sensitive lemmatization, Bergmanis & Goldwater (2018, p.1391) states that the context contains useful information beyond the wordform that helps lemmatizing unseen words. Bergmanis and Goldwater (2019, p.1) has argued in a recently written paper (published online in July 2019) that “lemmatization in context can improve accuracy on ambiguous and unseen words, provided the context sensitive lemmatization trains on complete sentences labeled with POS and/or morphological tags as well as lemmas, and have only been tested with 20k-300k training tokens”.

Lexical ambiguity is the one of the major motivation for context-sensitive lemmatization because quite often lemmatizers have to depend on context (Bergmanis & Goldwater, 2018). Relying on the context, Freiha and his colleagues (2018) have presented a lemmatization tool that is composed of the fusion of a machine-learning-based classifier (as a main lemmatizer) and of an auxiliary dictionary-based lemmatizer, with the underlying idea that is kind of lemmatization tool is well-suited to solve the cases of lexical ambiguity while the dictionary-based extension provides an extra performance boost.

The difficulty of the lemmatization task largely depends on several factors such as morphological productivity, lexical ambiguity and morphological regularity, morphophonological rules of a language or other phenomena such as vowel harmony or spelling changes and hence is likely affect its accuracy (Bergmanis & Goldwater, 2018). Ingason *et al.* (2009) argue that the morphological richness affects lemmatization and the corresponding large tagset also affect the accuracy of the parts of speech tagging. The pattern across languages emerge that the success of data-driven lemmatization depends on a language’s productivity, ambiguity, and regularity and accuracy tends to be higher for languages with low productivity while accuracy tends to be higher for languages with high ambiguity (Bergmanis & Goldwater, 2018, p.1399).

While adopting a simple data-driven context-sensitive approach to lemmatizing word forms in running text and treating lemmatization as a classification task for machine learning, Chrupała (2006) presented a method that automatically induce class labels by computing a Shortest Edit Script (SES) between reversed input and output strings. Commenting on the work of Chrupała (2006), Barteld et al. (2016) expresses the opinion that it is a sequence labeling approach to lemmatization in which a token is labeled with a rule that transforms it to its lemma and the set of rules from which the labels are chosen are induced automatically from the training data.

Chrupała *et al.* (2008, p.2362) had suggested the Morfette system which is “composed of two learning modules, one for morphological tagging and one for lemmatization, and one decoding module which searches for the best sequence of pairs of morphological tags and lemmas for an input sequence of wordforms”. Morfette relies on the concept of edit trees and

a simple perceptron is used for classification with hand-crafted features (Malaviya *et al.*, 2019). Some researchers believe that the edit tree based approaches can be very effective for highly synthetic languages (Manjavacas *et al.* 2019, p.10). Chrupala *et al.* (2008) have argued that errors which mostly affect unknown words could be dealt with successfully by (i) providing more training data, (ii) incorporating language specific resources such as gazeteers or lexicons into the model.

Muller *et al.* (2015) has made a significant contribution in the area of lemmatization by presenting the first joint log-linear model of morphological analysis and lemmatization, called LEMMING (a modular lemmatization model), that operates at the token level combining a wide variety of features of previous models and is also able to lemmatize unknown forms. LEMMING is available under an open-source licence (<http://cistern.cis.lmu.de/lemming>).

Chakrabarty *et al.* (2017, p.1481) presented a supervised, language independent, context sensitive lemmatization model with two-stage bidirectional gated recurrent neural network (BGRNN) architecture that needs *lemma* tagged continuous text to learn and the authors claim that the two most important advantages of this model are:

- (i) it is not necessary to define hand-crafted features such as the word form, presence of special characters, character alignments, surrounding words etc.
- (ii) the parts of speech and other morphological attributes of the surface words are not required for joint learning.

Lematus proposed by Bergmanis and Goldwater (2018) is a neural sequence-to-sequence model built using the Nematus machine translation toolkits which takes as input a character sequence representing the wordform in its N-character context, and outputs the characters of the lemma. In a recently written paper, Bergmanis and Goldwater (2019), reported how they improved the accuracy of their lemmatization model on ten languages both in low (1k) and medium (10k) resource settings, by a training data augmentation method that combines the efficiency of type-based learning and the expressive power of a context-sensitive lemmatization model.

Manjavacas *et al.* (2019, p.10) presented a method to improve lemmatization with encoder-decoder models by improving context representations with a joint bidirectional language modeling loss that sets a new state-of-the-art for lemmatization of historical languages and is competitive on standard languages. The authors believe that their joint language modeling loss which does not rely on any additional annotation, can be crucial in low resource and non-standard situations where annotation is costly.

With the aim to design a lemmatization model that best extracts the morpho-syntax from the sentential context, Malaviya *et al.* (2019), presented a simple joint neural model for lemmatization and morphological tagging that achieves state-of-the-art results on 20 languages from the Universal Dependencies corpora. The authors suggest that their joint morphological tagging and lemmatization is especially helpful in low-resource lemmatization and languages that display a larger degree of morphological complexity, but also performs well with morphologically rich languages.

In a recently published paper, Chaudhary & colleagues (2019, p.1) presented a hierarchical neural model for contextual morphological analysis with a shared encoder and independent decoders for each coarse-grained feature, to address the issues of both data sparsity and



having a tractable computation time. They proposed a two multilingual transfer approaches to address the issue of data scarcity, where they train on a group of typologically related languages and find that language-groups with shallower time-depths (i.e., period of time during which languages diverged to become independent) tend to benefit the most from transfer.

Balakrishnan and Llyod-Yemoh (2014, p.175), both stemming and lemmatization play a crucial role in increasing relevance and recall capabilities of a retrieval system and the number of indexes are reduced, when these techniques are used. As explained in the earlier paragraphs, stemming procedure is similar, but not identical to lemmatization. Stemming is a process to reduce words with the same stem to a common form whereas lemmatization removes inflectional endings and returns the base or dictionary form of a word (Balakrishnan and Llyod-Yemoh 2014, p.174). Further, the limitations of stemming include (i) there is no guarantee of a stem to be a legitimate word form and (ii) the words are considered in isolation (Chakrabarty *et al.*, 2017).

When we compare accuracy of stemming and lemmatization, stemmers pose a series of issues due to their design, which is based on general rules rather than on linguistic knowledge but lemmatizers don't have these problems in stemming process words are stemmed to artificial words instead of regular words whereas lemmatization connects every word to its lemma, which is another regular word, making it a versatile and end-to-end tool<sup>19</sup>. Further, developing a stemmer is far simpler than building a lemmatizer, because in the latter, deep linguistics knowledge is required to create the dictionaries that allow the algorithm to look for the proper form of the word<sup>20</sup>.

Balakrishnan and Llyod-Yemoh (2014), conducted a comparative study of stemming and lemmatization for document retrieval precision performances with a baseline ranking algorithm (i.e. with no language processing). They developed a search engine and the algorithms were tested based on a test collection. The authors reported that both mean average precisions and histograms indicate stemming and lemmatization to outperform the baseline algorithm. Further they found in their research that for the language modeling techniques, lemmatization produced better precision compared to stemming, although the differences are insignificant and the overall the findings suggest that language modeling techniques improves document retrieval, with lemmatization technique producing the best result.

Manjavacas *et al.* (2019) argues that while lemmatization is considered to be solved for resource rich languages such as English, it remains a challenge for morphologically complex (e.g. Estonian, Latvian) and low-resource languages with unstable orthography (e.g. historical languages). Finally, Chaudhary & colleagues (2019) have argued that most languages being under resourced, often exhibiting diverse linguistic phenomena and data scarcity, existing state-of-the-art models for languages have coupled deep learning with cross-lingual transfer learning to successfully tackle these challenges.

This chapter gives a comprehensive literature and background study needed to understand the research methodology and tasks to be achieved explained in the next chapter. Also understanding the current literature and works in the field of semantic similarity will be a useful step to carry forward this research. The gold standard data collected in this research

<sup>19</sup> [https://cdn2.hubspot.net/hubfs/2396105/Benchmarks/BITEXT\\_Lemmatization\\_benchmark.pdf](https://cdn2.hubspot.net/hubfs/2396105/Benchmarks/BITEXT_Lemmatization_benchmark.pdf)

<sup>20</sup> <https://blog.bitext.com/what-is-the-difference-between-stemming-and-lemmatization/>

can be applied to the various semantic similarity methods explained above and various customizations can be done to them in order to create customized models for various semantic similarity calculations in the HR domain which will be the main focus of this research. Chapter 3 will explain the research methodology carried out to obtain the gold standard data and the method used to verify its quality. Chapter 4 will explain the data collection and analysis process undertaken in this research followed by the obtained results in Chapter 5. Lastly, Chapter 6 will give a brief conclusion along with limitations of this research and the future work that can be carried out in order to improve the results achieved.



# Chapter 3

## Research Methodology

To identify a suitable research method is one of the primary tasks of the researcher. The choice of appropriate method is guided by the research topic, the research questions and the advantages and disadvantages of different research methods suitable for this research. The reason for choosing the appropriate research methodology that will enable me to answer the research questions is discussed in this section.

Questionnaire survey is a well-recognized and commonly used technique within social science research for gathering information on participant social characteristics, present and past behaviour, standards of behaviour or attitudes and their beliefs related to the topic under investigation (Bird, 2009). Questionnaire survey allows the researcher(s) to quickly and efficiently collect data (Hewitt *et al.* 2017) and some of the major advantages of questionnaire survey include<sup>21</sup>:

1. Large amounts of information can be collected from a large number of people in a short period of time and at a reasonable cost
2. It can be carried out by the researcher with limited affect to its validity and reliability
3. The results of the questionnaires can usually be quickly and easily quantified by either a researcher or through the use of a software package
4. It can be analysed scientifically and objectively
5. When data has been quantified, it can be used to compare and contrast other research and may be used to measure change

The basic process of developing a questionnaire survey can be outlined as follows (Burgess, 2001):

1. Define the research aims.
2. Identify the population and sample.
3. Decide how to collect replies.
4. Design the questionnaire.
5. Run a pilot survey.
6. Carry out the main survey.
7. Collect the data.

Hewitt (et al. 2017) adds critical evaluation to this list and argues that if a critical evaluation is carried out by research team or a group of stakeholders, then it allows us to understand whether we actually received the information we expected, and, if not, how we might modify the questionnaire in the future.

Since completing a questionnaire is a cognitive burden for respondents because they are required to read the questions, think intently, and respond, Harlacher (2016) argues that to enable respondents to answer accurately, the questionnaire should be framed in a simple language to get accurate results from the survey. To create an efficient questionnaire with a lower cognitive burden, Harlacher (2016) suggests the researchers may consider the following guidelines:

<sup>21</sup> [https://www.le.ac.uk/oerresources/lill/fdmvco/module9/page\\_51.htm](https://www.le.ac.uk/oerresources/lill/fdmvco/module9/page_51.htm) (accessed on 19 August 2019)

- 1) Avoid demanding or time-consuming questions, such as asking respondents to rank order 15 items or to read extensive text.
- 2) Define terms to reduce misinterpretation and the need for respondents to develop their own definitions.
- 3) Group questions on similar topics so respondents do not have to jump mentally among topics.
- 4) Provide checkboxes for responses instead of asking respondents to type or write responses. For example, ask respondents to check answers (for example: male, female) instead of writing the answer.
- 5) Place instructions where they are needed, not just at the beginning of the questionnaire, so that respondents do not have to recall the directions.
- 6) For online questionnaires, avoid questions that require scrolling or switching screens.

In addition to above, Bird (2009) emphasizes that it is necessary to sequence the questions in a logical order in the questionnaire to ensure that participants understand the purpose of the research and allowing them to answer the questions carefully. It could be a good idea, to initially carry out a pilot study to test the questionnaire, by administering the questionnaires face-to-face to a small sample of participants. The researcher(s) could determine whether or not participants are comfortable with the sequence, structure of questions, questionnaire length and determine if there were any other questionnaire design defects (Bird, 2009), which should be corrected before the questionnaire survey for data collection. Questionnaire survey has been selected as one of the methods to collect data because it is one of the most efficient, reliable and time-tested method of collecting data in research.

In this research, there were two questionnaires created to validate performance on same set of sentences presented to human annotators in different manner. In the first questionnaire, each question has a pair of randomly sampled sentences with responses of yes/no to be selected for answering whether the given pair of sentences are similar or not. In second questionnaire, the sentences were presented together as a single group and the idea was to cluster sentences that similar to one another. The reason behind making two questionnaires is to validate human performance in answering the questionnaires and to check whether they perform consistently. Since we are providing the same set of sentences in different manner the main goal is to comprehend how consistent is human response in both questionnaires.

The two types of online questionnaire surveys that have been administered in this research are described as below:

1. **Yes/No questionnaire:** In this questionnaire, human annotators needed to mark YES/NO based on the similarity in meanings of sentences. They were given a set of 20 sentences. Based on whether they think 2 sentences are similar or not based on their meanings, they had to mark yes or no. The general theme of the sentences in these questionnaires is "Increase females in management". The most important criterion for marking yes or no is that the sentences should be semantically similar to each other in order to be placed under the same group. As an example, "Sufficient advertisement for the new campaign" and "Proper advertisement of the campaign" are two sentences that should come under the same group as these two sentences are very similar to each other. Therefore, they would need to mark option "YES" in the questionnaire. However, "Allow flexible work hours" is not semantically similar i.e. meaning wise similar to the previous two sentences and hence should be marked as "NO" in the questionnaire.

2. **Grouping questionnaire:** In this questionnaire, human annotators needed to cluster texts based on the similarity in their themes. They were given a set of 20 different sentences. Based on what they think are similar sentences, they needed to cluster them in multiple clusters. There were no restrictions on the number of clusters that should be created but all sentences under a cluster should be related to the same topic i.e. the sentences under a cluster should be semantically similar to each other. The general theme of the sentences in this questionnaire is "Increase females in management". However, there can be various subgroups that can be made under the general theme based on how similar two sentences are. They did not need to name any clusters by the topic of the sentences under them but must make sure that the sentences under each group are similar to each other. Also, an important criterion for the clustering was that the sentences should be semantically similar to each in order to be placed under the same cluster. It was also necessary to mention that if found a sentence does not belong to any cluster, they could freely leave it separate and not forcefully add the sentence they find semantically dissimilar to a cluster.  
As an example, "Sufficient advertisement for the new campaign" and "Proper advertisement of the campaign" are two sentences that should come under the same cluster as these two sentences are very similar to each other. Both of them would come under the group category "increasing females in management by carrying out proper advertisement for the campaign".

Initially, a pilot study was conducted with 100 sentences for grouping given to the HR-domain expert and human annotators from the HR department. This was done to test whether they could successfully carry out the grouping task with such a large data set given to them at the same time. There were discrepancies seen in these clustering as they lost track of the number of groups formed and the sentences being assigned to these clusters because a large number of clusters were formed. It was also time consuming to answer this questionnaire. Along with that, it was also time consuming to create the yes/no questionnaire as the pair of sentences were randomly sampled from the large set of sentences which made it difficult to keep a track of whether the sentences were correctly represented in the questionnaire. Hence, the decision was made to create the questionnaires with smaller data sets so that the above-mentioned problems can be controlled to some extent.

The grouping questionnaires for the creating gold standard data set were given to the HR-domain expert. The questionnaires each had a set of 20 sentences only. He performed clustering on the sentences to group them based on their semantic similarity. Since the HR-domain expert has years of experience in this domain, his grouping of sentences based on semantic similarity was considered as the gold standard data set for comparison with other human annotators. These benchmarks will include the best groupings of text objects which will be treated as the ground truth for evaluation. This should be helpful as domain-experts are expected to have the necessary knowledge which will give me a benchmark for comparison.

There were five other human annotators from the human resources department who were given both types of questionnaires i.e. grouping questionnaire as well as yes/no questionnaire to perform clustering of the sentences based on their knowledge of the HR domain. As there were five sets of 20 sentences each, the grouping and yes/no questionnaire were made from each of these set of sentences. Therefore, there were in total 5 grouping questionnaires and 5 yes/no questionnaires that were answered by the human annotators.

The design of the questionnaires was done in such a manner so as to aid in answering the research questions mentioned in Section 1.1. The two different types of questionnaires were formed so as to check whether there is a consistency between the annotators' answers when presented in two different ways i.e. as yes/no questionnaire and individual clustering questionnaire. Also, the gold standard data given to the HR-domain expert was verified by giving the individual clustering questionnaire to other human annotators. This was done to check whether the results were reproducible and check the quality of the gold standard data for HR domain. The hypothesis was that there should be high level of agreement in the clustering results between the gold standard data and human annotators' clustering results.

The yes/no questionnaire consisted of 20 questions each. In every question, a randomly sampled pair of sentences were given to them. As the number of combinations that could be made out of 20 sentences from the data set was large, therefore the decision was made to only select a random pair of 20 sentences to be given to the human annotators. For example, "Are sentences "A parity friendly corporate culture" and "Making top management aware of the current gender parity issues" similar based on their meaning?" question was given to them. This question in the questionnaire had two options YES or NO between which they had to choose based on whether they think that these two sentences have a similar meaning or not. The above mentioned and similar questions were given over the five different questionnaires based on different data sets (each consisting of 20 sentences) collected during the brain storming sessions of the HR department.

The grouping questionnaire consisted of giving the human annotators all the 20 sentences together at the same time. They had to group them together based on how similar they thought the meanings of the sentences were. The idea here was that reading all the sentences at the same time might give them a better idea of general theme of the sentences which could help them to better cluster the sentences in the proper groups. For example, the list of below mentioned sentences were given to them at the same time. They had to perform grouping from this list of sentences and place all the sentences they find to be semantically similar i.e. sentences whose meanings they find to be similar in the same cluster.

1. A parity friendly corporate culture
2. Coordination by the planning team
3. A presentation outlining the benefits of gender parity
4. Generation of a financial estimation of potential benefits
5. Focus on lowering the attrition rate
6. Creating parity issues awareness for top management
7. Advertise the campaign
8. Dedicated presentation of the gender parity benefits
9. Corporate support for parity
10. Having a team member realizing a financial estimation of the benefits
11. Top Management support to the new policy
12. Offer flexible working arrangements
13. A communication professional involved on the communication strategy and benefits
14. A lawyer stating the emerging compliance issues concerning gender parity
15. Dedicated head-hunter team
16. Exposing women to all company operations and functions
17. Identifying women to be involved in cross-functional projects
18. Making top management aware of the current gender parity issues
19. Legal analysis regarding compliance issues of gender parity
20. Create a recruitment team

It was also made sure that the human annotators are not given the two-different kind of questionnaires from the same sentences data set at the same time. There was some gap between the time span when the questionnaires were presented so that they did not remember the sentences or as a matter of fact any of the answers from the previously answered questionnaire. This was done to make sure that there are no biases in answering the two-different kind off questionnaire as every data set had two questionnaires (yes/no and individual clustering) made out of it.

The final evaluation was carried out based on comparison between grouping done by human annotators from the HR department with the set of domain specific benchmarks, that was developed with the help of the HR domain-expert. Finally, to analyse agreement on sentence pair classification between gold standard data and individual clustering as well as gold standard data and sentence pair data, F1 score, precision and recall values were calculated which will be described further in the next chapter.

# Chapter 4

## Data Collection and Analysis

Gold standards denote scientific procedures or collections which are accepted standard process (Wissler *et al.*, 2014). They have been initially used to support the evaluation of the interaction between semantic distance measures and of linguistic and knowledge resources (Barzegar *et al.* 2018). Gold standard corpora in NLP context are manually annotated collections of text and for high quality gold standard corpora multiple experts view the data independently and the inter-annotator agreement is computed to ensure quality, which makes the creation of gold standard corpora a very costly process (Wissler *et al.*, 2014).

To support Natural language processing (NLP) studies, modeling, supervised machine learning, and testing typically occur at various stages of the analyses and these tasks are facilitated by utilizing a gold standard corpus containing annotations that adequately represent the concepts contained in the domain under analysis (Juckett, 2012). Baroni & Lenci (2011) argue that a data set intending to represent a gold standard for evaluation should include tests items that are as little controversial as possible. However, some gold standard such as WordNet is biased and hence also lacks domain specific sense definitions while providing an abundance of sense definitions that occur too rarely in most corpora (Bordag, 2006).

Often, the size of the gold standard corpus appears to be determined by ad hoc procedures that are constrained by financial and personnel resources rather than by statistical sampling procedures and in many reports, the number of documents in the gold standard corpus is simply stated without rationale (Juckett, 2012). Hence Bordag (2006) who used a triplet-based hierarchical graph clustering approach, suggested four measures (Biemann, 2006):

1. Retrieval precision (rP): similarity of the found sense with the gold standard sense
2. Retrieval recall (rR): amount of words that have been correctly assigned to the gold standard sense
3. Precision (P): fraction of correctly found disambiguations
4. Recall (R): fraction of correctly found senses

Like most existing approaches, Bordag's method utilizes clustering of word co-occurrences, but this approach differs from other approaches to word sense induction<sup>22</sup> (WSI) in that it enhances the effect of the one sense per collocation observation by using triplets of words instead of pairs (Bordag, 2006).

Juckett (2012) suggests that the estimates for the acceptable size of a gold standard corpus are derived from the probabilities of capturing target words from a working corpus during random sampling and provides an overview of the procedures to determine the number of documents needed for a gold standard corpus, by following the steps mentioned below:

1. Pre-select a working corpus from clinical text documents such that it meets the needs of the study and suitably represents the population of interest.

<sup>22</sup> The aim of word sense induction (WSI) is to find senses of a given target word automatically and if possible in an unsupervised manner and it is akin to word sense disambiguation (WSD) both in methods employed and in problems encountered, such as vagueness of sense distinctions (Bordag, 2006).

2. Select a comparative corpus containing appropriate common word usage.
3. Convert the clinical text and the comparative corpus text into word tokens.
4. Subtract the set of comparison text tokens from the set of clinical text tokens creating a remainder set.
5. Compute frequencies of token occurrences in the remainder set.
6. Calculate the capture probabilities for each unique token as a function of various choices for gold standard corpus size. Integrate (weighted sum) the probabilities into one value for each corpus size.
7. Select a corpus size depending on an acceptable capture probability.

The above steps give a general strategy to determine the number of documents needed for a gold standard corpus which is a representative sample of all documents (Juckett, 2012). Juckett (2012) has demonstrated a method for calculating the size of a gold standard corpus which is dependent on choices for capture probabilities, comparison corpora, and word length selection.

In relation with the multiplicity of gold standards, there is question of whether the performance of a language processing system should be measured against a theoretical objective (the maximal performance value defined by the evaluation metrics), or rather against the average performance level displayed by humans when performing the task under consideration (Paroubek *et al.*, 2007).

Juckett (2012) expresses the opinion that to obtain a representative sample for a gold standard, it is incumbent on a researcher to examine as many variables as possible between the sample corpus and the total corpus to justify the final product and he points out that the creation of a gold standard corpus for use in NLP modeling, testing, and machine learning is expensive, time consuming, and requires specialized personnel (Juckett, 2012). Filannino & Di Bari (2015) has argued that collecting and manually annotating gold standards in NLP has become so expensive that in the last years the question of whether we can satisfactorily replace them with automatically annotated data (silver standards) is arising more and more interest.

The data for this research was collected during the multiple brainstorming sessions on the theme “increase females in management” of the HR department. During these sessions, they would discuss on a more focussed topic within this general theme and people would suggest their ideas of how females can be increased in management. A complete compilation has been made from the ideas of people which has been used for measuring semantic similarity in this research. The data set comprises of 200 short sentences. The main theme of the sentences is “increase females in management”. Some examples of the sentences are as follows:

- Assigning women managers visible and challenging tasks
- Increase women's salaries for positions showing disparity
- Financial analysis of payroll parity measures
- Communicating new values regarding gender parity
- Top-management involvement for gender equality values communicated by internal and external means of communication
- Recruiting more women
- Launch external recruitment campaign
- Launch Internal recruitment campaign
- Offer Flexible working arrangements

- Identify departments/teams to recruit for

The above mentioned and other similar sentences were then used to create the two types of questionnaires. The grouping questionnaire was given to the HR-domain expert who performed clustering on the sentences which has been treated as the gold standard data for this research. The other human annotators were given both the grouping as well as yes/no questionnaire as a measure of comparison with the gold standard data.

Clustering of sentences has been used as a measure for semantic similarity as there are several advantages of using clustering for semantic similarity evaluation of sentences. Clustering is a key technique in pattern recognition, data mining, and knowledge discovery where the aim is to uncover the (hidden) structure underlying a given collection of objects (Bouchachia, 2012). One of the major advantages of clustering is that it provides more information than a single context, particularly where corpus analysis and any kind of statistical processing is involved (Maynard & Ananiadou, 1999).

Clustering is a better method for the semantic similarity evaluation of sentences because of the reasons given below (Naik *et al.* 2015; Saiyad *et al.* 2016):

- 1) Clustering is the method to make groups of documents on the basis of their conceptual similarity therefore, it makes the task easier while working with unknown collection of unstructured text.
- 2) Clustering helps in information relationship discovery.
- 3) Semantic approach helps in information and relationship discovery among terms of the documents.
- 4) Clustering helps in retrieving the relevant data according to user query.
- 5) Clustering can help in semantically relating the clusters to one another.
- 6) In particular, Latent Semantic Indexing (LSI) technique can achieve dynamic clustering on the basis of conceptual contents of documents. LSI can carry out example based categorization as well as cross linguistic concept searching. LSI can also process random character strings. This technique is not limited to work only with words. It is proven that LSI is good solution for a number of conceptual matching problems. This technique can capture key relationship information containing casual information, goal oriented and taxonomic information.
- 7) Semantic information retrieval method has exploited the advantages of the semantic web to retrieve the relevant data.
- 8) Semantic information is used for improving evaluation measures like precision or recall in information retrieval system and clustering process.

Further clustering is a useful form of knowledge acquisition tool because it enables us to make use of information about clusters to make sense of the context where individual words otherwise provide incomplete information (Maynard & Ananiadou, 1999).

After data collection, the most challenging task facing the researcher is to reach across multiple data sources (Miles and Huberman, 1994) and to compare, contrast, corroborate and put together an analysis of the data in a meaningful way. Ideally the researcher(s) should set aside opinions and allow the analysis to be data driven (Percy et al. 2015). Data analysis is essentially a reflective process that enables the researcher to develop an understanding of the phenomenon under investigation (Moustakas, 1994). Stake (2000, p. 445) argues that ‘in being reflective the researcher is committed to pondering the impressions, deliberating recollections and records’. The reflective process ‘provides a logical, systematic, and coherent’ means to carry out the analysis needed to arrive at a full description of the



phenomenon under study (Moustakas, 1994, p 47). The data collected from participants (questionnaires) are first analysed individually and then repeating patterns/themes from all participants are synthesized together into a composite synthesis, which attempts to interpret the meanings/implications regarding the topic under investigation (Percy et al. 2015).

The final evaluation consists of measuring and analysing agreement on clustering done by the gold standard data and individual clustering as well as gold standard and sentence pair classification. For this measurement, F-score, precision and recall have been used to understand the level of agreement.

F-score is one of the most commonly used measures in Information Retrieval, Natural Language Processing and Machine Learning (Powers, 2015). It is derived from two summary measures: precision and recall; while precision describes the frequency of retrieved documents which a system returns that are correct (Derczynski, 2016), whereas recall is the frequency with which relevant documents are retrieved or ‘recalled’ by a system (Powers, 2015). Although van Rijsbergen did not define the formula of the F-score per se, it is widely accepted that the origin of the definition of the F-score is traced to van Rijsbergen’s E (effectiveness) function (van Rijsbergen, 1979, Sasaki, 2007).

F-score is based on the confusion matrix as shown in Table 2 below.

**Table 2: Confusion matrix**

| <b>Class / Recognized</b> | <b>as Positive</b> | <b>as Negative</b> |
|---------------------------|--------------------|--------------------|
| <b>Positive</b>           | <i>tp</i>          | <i>fn</i>          |
| <b>Negative</b>           | <i>fp</i>          | <i>tn</i>          |

The confusion matrix can be used to define precision P and recall (or sensitivity) R as:

$$P = \frac{tp}{tp + fp}$$

$$R = \frac{tp}{tp + fn}$$

The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. The recall is intuitively the ability of the classifier to find all the positive samples. In practice F-score is then represented in an equation shown below as a harmonic weighted mean of precision and recall (Derczynski, 2016).

$$F_{\beta} = \frac{(\beta^2 + 1)PR}{\beta^2P + R}$$

In the above equation, ‘F’ is F score, ‘P’ is Precision, the probability that a randomly chosen predicted instance (positive) will be relevant and ‘R’ is Recall, the probability that a randomly chosen relevant instance will be predicted (positive) (Powers, 2015). The coefficient  $\beta$  is a parameter that controls a balance between P and R. If  $\beta > 1$ , F becomes more recall-oriented and if  $\beta < 1$ , it becomes more precision oriented (Sasaki, 2007). When  $\beta = 1$ , F-score becomes equivalent to the harmonic mean of P and R and it is called “F1 score” (Derczynski, 2016). F-score is intended to combine these into a single measure of search ‘effectiveness’ (Powers, 2015).

The metric has evolved to be used for three different averages, namely micro, macro and weighted.

- Let  $y$  be the set of predicted (sample, label) pairs
- $y^{\wedge}$  be the set of true or gold standard (sample, label) pairs
- $L$  the set of labels
- $y_l$  the subset of  $y$  with label  $l$ , or formally:  $y_l := \{(s, l') \in y \mid l' = l\}$
- $P(A, B) := \frac{|A \cap B|}{|A|}$
- $R(A, B) := \frac{|A \cap B|}{|B|}$  (where  $R(A, B) := 0$  and  $P(A, B) := 0$  for  $B = \emptyset$ ), and
- $F_{\beta}(A, B) := (1 + \beta^2) \frac{P(A, B) \times R(A, B)}{\beta^2 P(A, B) + R(A, B)}$

Then the metrics are defined as given below:

$$F_{\beta\text{-micro}} = F_{\beta}(y, \hat{y})$$

$$F_{\beta\text{-macro}} = \frac{1}{|L|} \sum_{l \in L} F_{\beta}(y_l, \hat{y}_l)$$

$$F_{\beta\text{-weighted}} = \frac{1}{\sum_{l \in L} |\hat{y}_l|} \sum_{l \in L} |\hat{y}_l| F_{\beta}(y_l, \hat{y}_l)$$

Many NLP systems are evaluated using F-score, which describes system performance using a scale from zero to one (Derczynski, 2016). Alguliev & Aliguliyev (2008) used cosine measure and F-score to calculate similarity between sentences, and after comprehensive experimental evaluation, they came to the conclusion that F-score lead to the best overall results than cosine measure.

However, critics of F-score have argued that “it is based on a mistake, and the flawed assumptions render it unsuitable for use in most contexts” (Powers, 2015). Derczynski (2016) points out that some of the disadvantages of F-score include it lacks detail, it is unable to distinguish low-recall from low-precision systems and two systems may reach the same F-score, but on very different examples, depending on what information they use. In addition, Powers (2015) expresses the opinion that F-score focuses on one class only, it is biased by the majority class and as a probability assumes the real and prediction distributions are identical. Further comparing F-score differences don’t tell us how different the mistakes made by each classifier are, and therefore sheds only minimal light on how helpful classifier combination might be (Derczynski, 2016).

To overcome the shortcomings of F-score, Derczynski (2016) proposed the adaptation of complementary precision, recall and F-score in situations where the differences between system outputs is of help in analysis; particularly in shared evaluation tasks, in feature ablation, and in cases where traditional F-score results are very close. Powers (2015) suggest that weighting F-score simply by the size of each class (or the number of predictions of each class) enshrines a bias when these are different, and means a better result can be achieved by changing the bias towards the more prevalent classes (and some learning algorithms do this).

The program for evaluation of sentence pair classification agreement is written in Python as it is the default choice of programming language for Natural Language Processing tasks. Python is perceived as an object-oriented language. In this thesis, the implementation by Scikit-learn version 0.21.3 is used where possible. There is no need to define any new classes

instead the classes “precision\_recall\_fscore\_support” and “classification\_report” are imported to perform the tasks.

The first code is written to take input from the multiple csv files which contain the gold standard data and results from the questionnaires answered by human annotators. The input is then cleaned in the format:

<sentence group number>

by removing tabs and initial sequence numbering present in the csv files to make them suitable for processing. Finally, a function is created which has a dictionary called “classes” that stores the cluster numbers representing each sentences’ group number. This is done individually for all gold standard data sets, sentence pair classification data sets and individual clustering data sets.

The second code is written with the help of scikit-learn which is used to calculate precision, recall and F-score between gold standard and sentence pair classification data as well as gold standard and individual clustering data. The files generated from the previous pre-processing have been then used as input to this code.

The sentence pair classification data sets, i.e. data from yes/no questionnaire, have been compared as a binary classification task with gold standard. In the script, when the input values are given, “0” signifies that the sentences have been clustered in the same group by the human annotator but the gold standard clusters in a different group based on semantic similarity and vice-versa. Similarly, “1” signifies that there is match in the way human annotator and gold standard have classified the pair of sentences i.e. they have both classified the sentences to be in the same cluster or in the different clusters based on their semantic similarity.

The grouping questionnaire i.e. data from individual clusters by human annotators have been compared as a multi class classification task with gold standard. In the script, the classes from five different questionnaires have been input and compared individually with the clusters in the gold standard data.

Finally, the average for precision, recall and F-score is calculated for all the yes/no questionnaires as well as individual clustering questionnaires. It is expected that the larger the F Score is, the better is the clustering performance and agreement with the gold standard data.

# Chapter 5

## Results

This section presents the results as obtained from the analysis from the Python code explained above. The main goal here was to find out on average which questionnaire amongst the yes/no questionnaire and individual clustering questionnaire had the most agreement with the gold standard. The table given below shows the average F-score, average precision and average recall for the comparison between gold standard and yes/no questionnaire as well as gold standard and individual clustering.

**Table 3: Results table depicting average precision, average recall and average F-score**

| Questionnaire         | Average Precision | Average Recall | Average F-Score |
|-----------------------|-------------------|----------------|-----------------|
| Individual Clustering | 0.37              | 0.36           | 0.31            |
| Yes/No                | 0.62              | 0.56           | 0.57            |

It was expected that the individual clustering would give the most similar final clustering results with gold standard as compared to yes/no questionnaire. The reason behind this is the human annotators could read all the sentences at the same time and thus would be able to interpret better thus leading to correct clustering. However, that was not the case and yes/no questionnaire was most similar to the gold standard data.

**Table 4: Results table with precision, recall and F-score for all annotators for different questionnaires**

| Questionnaire            | Average Precision | Average Recall | Average F-Score |
|--------------------------|-------------------|----------------|-----------------|
| Individual Clustering Q1 | 0.58              | 0.40           | 0.39            |
| Individual Clustering Q2 | 0.31              | 0.35           | 0.31            |
| Individual Clustering Q3 | 0.26              | 0.30           | 0.22            |
| Individual Clustering Q4 | 0.19              | 0.35           | 0.25            |
| Individual Clustering Q5 | 0.49              | 0.40           | 0.35            |
| Yes/No Q1                | 0.54              | 0.60           | 0.57            |
| Yes/No Q2                | 1.00              | 0.33           | 0.50            |
| Yes/No Q3                | 0.64              | 0.69           | 0.66            |
| Yes/No Q4                | 0.40              | 0.54           | 0.46            |
| Yes/No Q5                | 0.55              | 0.65           | 0.59            |

In the calculation of F-score, it is usually expected that the F-score value is between precision value and recall value. However, in the sentence agreement program while calculating precision, recall and f-score using scikit-learn's `precision_recall_fscore_support`, the average parameter's value was taken to be weighted average. This was done to take into account label imbalance. The "weighted" option calculates metrics for each label and finds the average weighted by support i.e. the number of true instances for each label. The support is the total number of occurrences of every class in `y_true`. This resulted in an F-score that is not between precision and recall.

Also, in the questionnaire, human annotators were asked to strictly perform the grouping based on semantic similarity of the sentences. It was mentioned specifically that the clustering should be done on how similar the sentences are based on their meaning instead of lexical similarity or word-to-word similarity. However, during data analysis, it was found that some human annotators have performed the clustering not just on the basis of semantic similarity but also rather word-to-word similarity. If they found that sentences had the exact same words, they would be more inclined to put them in the same group immediately instead of taking time to think over the meaning or the ideas the sentences were trying to convey. For example, some annotators said yes to placing “Recruit more female candidates” and “List of eligible female candidates” whereas according to gold standard data these two sentences should be placed in separate groups as one of them involves recruitment process and the other one is making a list of eligible female employees. Some human annotators placed them in the same cluster because they found the same group of words “female candidates”. This was found to happen more in the yes/no questionnaires as compared to the individual clustering questionnaires. The reason behind this could be that in the individual clustering questionnaire they could read and understand the general idea behind all the sentences at the same time whereas that was not possible in the yes/no questionnaire because they could only see the pair of sentences presented to them. This helped them to judge and perform better in clustering the sentences.

# Chapter 6

## Conclusion

Experts express the opinion that Natural Language Processing community is yet found “the best supervised task for embedding the semantics of a whole sentence” (Conneau et al., 2017, p.671). Soğancıoğlu & colleagues (2017) argue that there is a need to develop domain-specific approaches in sentence semantic similarity measures using domain-specific corpora or knowledge sources, due to the huge amount of information available in textual format, which will make the retrieval, extraction and summarization of information vital more effective. In current research, not many data sets have been collected to create gold standard data specific to the HR domain. This thesis is aimed at taking a closer step towards this goal to develop and make available domain-specific corpora for further research. The results from this thesis can be utilized for further research on text semantic similarity in the HR domain. The gold standard data collected can be used to train models built for semantic similarity and give better results for tasks such as document collection classification specific to the HR-domain context. Also, as given in the literature review, applying stemming and lemmatization to the sentences when feeding them as inputs to various similarity measure algorithms can improve the results for sentence classification based on their similarities.

Another point to take note of is that the questionnaire was formed on a limited data set. Although it is a good stepping point towards our goal but it is not a sufficiently large data set to verify it under all circumstances. Hence, for future research the collection of a larger data set to create gold standard for understanding and training models for semantic similarity would be an interesting research. Also, for creation of questionnaires, methodologies other than random sampling should also be applied so that better representation of sentences can be done for the yes/no questionnaire keeping in mind all the possible combinations of sentences from the data set. Different types of questionnaires can also be administered to test the validity of the results.

Lastly, this research does not take into account cross-cultural differences which affect understanding of language. Cambria & White (2014) argue the NLP systems can process a database of millions of common-sense facts, but that is not enough for computational natural language understanding, what is required is to focus the research on developing the capability of the NLP systems to handle human knowledge, interpret emotions and cultural nuances. For example, people in the HR of a multinational company speak a common language. The words used to construct a sentence in the same language can be perceived differently by people from different cultural backgrounds. The development of cross-cultural language similarity would be an interesting further research to pursue in the HR-domain. Also, the development of cross-language similarity methods for similar purposes could be another interesting aspect for text semantic similarity. The cross-language similarity of two texts can be computed in the same way as the similarity of two texts in the same language, by using the similarity between the words. The cross-language similarity of two texts could be used in second language teaching to select similar texts, or in cross-language information retrieval.

# List of References

- Agarwal, N., Rawat, M. and Maheshwari, V. (2014), Comparative Analysis of Jaccard Coefficient and Cosine Similarity for Web Document Similarity Measure, *International Journal for Advance Research in Engineering and Technology*, Volume 2, Issue 10, pp.18-21.
- Aggarwal, M. (2011), Information Retrieval and Question Answering NLP Approach: An Artificial Intelligence Application, *International Journal of Soft Computing and Engineering*, ISSN: 2231-2307, Volume1, Issue-NCAI2011, June 2011.
- Alguliev, R. M. and Aliguliyev, R. M. (2007), Experimental investigating the F-measure as similarity measure for automatic text summarization, *Applied and Computational Mathematics*, Volume 6, No.2, pp.278-287.
- Andrews, S. and G. Hamarneh, G. (2015), Multi-region probabilistic dice similarity coefficient using the Aitchison distance and bipartite graph matching, arXiv preprint arXiv:1509.07244.
- Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, Richard Socher, Ask Me Anything: Dynamic Memory Networks for Natural Language Processing, *Proceedings of The 33rd International Conference on Machine Learning, PMLR 48:1378-1387*, 2016.
- Balakrishnan, V. and Llyod-Yemoh, E. (2014), Stemming and lemmatization: A comparison of retrieval performances, *Lecture Notes on Software Engineering*, Vol.2, pp. 262-267. DOI: [10.7763/LNSE.2014.V2.134](https://doi.org/10.7763/LNSE.2014.V2.134)
- Baroni, M., & Lenci, A. (2011). How we BLESSed distributional semantic evaluation. In Proceedings of the EMNLP GEMS Workshop, pp. 1–10, Edinburgh, UK.
- Barteld, F., Schroder, I. and Zinsmeister, H. (2016), Dealing with word-internal modification and spelling variation in data-driven lemmatization, *In Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pp. 52–62. Association for Computational Linguistics.
- Barzegar, S., Davis, B., Zarrouk, M., Handschuh, S. and Freitas, A. (2018), SemR-11: a multi-lingual gold-standard for semantic similarity and relatedness for eleven languages. *In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.
- Benedetti, F., Beneventano, D., Bergamaschi, S. and Simonini, G. (2019), Computing inter-document similarity with Context Semantic Analysis, *Information Systems*, Volume 80, February 2019, pp.136-147.
- Bergmanis, T. and Goldwater, S. (2018), Context sensitive neural lemmatization with lemmatus, *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1, pp.1391– 1400.
- Bergmanis, T. and Goldwater, S. (2019), Training Data Augmentation for Context-Sensitive Neural Lemmatization Using Inflection Tables and Raw Text, [arXiv:1904.01464v3](https://arxiv.org/abs/1904.01464v3) [cs.CL]. (accessed on 29 July 2019).
- Biemann, C. (2006), Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems, *Workshop on TextGraphs, at HLT-NAACL 2006*, pp. 73–80, New York City, June 2006, © 2006 Association for Computational Linguistics
- Bisandu, D. B., Prasad, R. and Liman, M (2019), Data clustering using efficient similarity measures, *Journal of Statistics and Management Systems*, Volume: 22 (4), pp.1-23.

- Bordag, S. (2006), Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation. Proceedings of EACL-06. Trento.
- Bornmann, L., Wray, K. B. and Haunschild, R. (2019), Citation concept analysis (CCA)-A new form of citation analysis revealing the usefulness of concepts for other researchers illustrated by two exemplary case studies....., arXiv preprint arXiv:1905.12410.
- Bouchachia, A. (2012), Dynamic Clustering, *Evolving Systems*, September 2012, Volume 3, Issue 3, pp. 133–134.
- Cambria, E. and White, B. (2014), Jumping NLP Curves: A Review of Natural Language Processing Research, *IEEE Computational Intelligence Magazine*, pp.48-57; May 2014.
- Cannon, B. J., Mikiyasu, N., Daisuke, S., and Ash, R. (2018), Shifting Policies in Conflict Arenas: A Cosine Similarity and Text Mining Analysis of Turkey’s Syria Policy: 2012-2016, *Journal of Strategic Security* Vol.11, No. 4, pp.1-19.
- Carlson, S. (2006), Challenging Google, Microsoft Unveils a Search Tool for Scholarly Articles, *Chronicle of Higher Education*, Volume 52 (33), p.A43.
- Cer, D., Diab, M., Agirre, E., Lopez-Gazpio I. and Specia L. (2017), SemEval-2017 Task 1: Semantic Textual Similarity-Multilingual and Cross-lingual Focused Evaluation, arXiv preprint arXiv:1708.00055 2017.
- Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., St. John, R., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., Sung, Y., Strophe, B. and Kurzweil, R (2018), Universal Sentence Encoder, arXiv:1803.11175 [cs.CL], 12 April 2018. (accessed on 27 June 2019)
- Chakrabarty, A., Pandit, O. A. and Garain, U. (2017), Context sensitive lemmatization using two successive bidirectional gated recurrent networks. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1481–1491, Vancouver, Canada, ©Association for Computational Linguistics. <https://doi.org/10.18653/v1/P17-1136>.
- Chaudhary, A., Salesky, E., Bhat, G., Mortensen, D. R., Carbonell, J. G. and Tsvetkov, Y. (2019), CMU-01 at the SIGMORPHON 2019 Shared Task on Crosslinguality and Context in Morphology. [arXiv:1907.10129v1](https://arxiv.org/abs/1907.10129v1) [cs.CL] (accessed on 30 July 2019).
- Chauhan, S. S. and Batra. S. (2018), A parallel computational approach for similarity search using Bloom filters, *Computational Intelligence*, Vol. 34; pp. 713–733.
- Chen, Q., Kim, S., Wilbur, J. and Z. Lu. (2018) Sentence similarity measures revisited: ranking sentences in PubMed documents. In *ACMBCB’ 18: 9th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, August 29-September 1, 2018, Washington, DC, USA., ACM, NY, NY, USA 2 pages.
- Chowdhury, G. (2003), Natural language processing, *Annual Review of Information Science and Technology*, Vol. 37, pp. 51-89, ISSN 0066-4200.
- Chrupała, G. (2006), Simple data-driven context sensitive lemmatization, *Procesamiento del Lenguaje Natural*, Vol. 37, pp.121–127.
- Chrupała, G., Dinu, G. and Genabith, J. V. (2008), Learning Morphology with Morfette, *In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, European Language Resources Association (ELRA), Marrakech, Morocco.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P. P. (2011), Natural Language Processing (almost) from Scratch, *Journal of Machine Learning Research*, Vol.12, pp.2493-2537.
- Conneau, A., Kiela, D., Schwenk, H., Barrault, L. and Bordes, A. (2017), Supervised learning of universal sentence representations from natural language inference data, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*,



- p.p.670–680, Copenhagen, Denmark, September 7–11, 2017. © Association for Computational Linguistics.
- Couto, F. M. and Lamurias A. (2019) Semantic Similarity Definition. In: Ranganathan, S., Gribskov, M., Nakai, K. and Schönbach, C. (eds.), *Encyclopedia of Bioinformatics and Computational Biology*, Vol. 1, pp. 870–876.
- Day, H. E. and Edelsbrunner, H. (1985), Efficient Algorithms for Agglomerative Hierarchical Clustering Methods, *Journal of Classification*, Volume 1, pp.7-24.
- Derczynski, L. (2016), Complementarity, F-score, and NLP Evaluation, *In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 23–28 May 2016.
- Dice, L. R. (1945), Measures of the Amount of Ecologic Association Between Species, *Ecology*, Vol. 26, No. 3, pp. 297-302.
- Dong, A. and Bhanu, B. (2003), Active Concept Learning for Image Retrieval in Dynamic Databases, *In Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)*, pp.1-6.
- E. Cambria and A. Hussain (2012), *Sentic Computing: Techniques, Tools, and Applications*. Dordrecht, The Netherlands: Springer-Verlag, pp.1-176.
- Erjavec, T. and Dzeroski, S. (2004), Machine learning of morphosyntactic structure: Lemmatizing unknown Slovene words, *Applied Artificial Intelligence*, Vol. 18, pp.17–41.
- Fagan, J. C. (2017), An Evidence-Based Review of Academic Web Search Engines, 2014-2016: Implications for Librarians' Practice and Research Agenda, *Information Technology and Libraries*, Vol. 36(2), pp.7-47.
- Filannino, M. and Di Bari, M. (2015), Gold standard vs. silver standard: the case of dependency parsing for Italian, *In: Proceedings of the Second Italian Conference on Computational Linguistics CLiC-it 2015*, 3-4 December 2015, Trento.
- Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G. and Ruppin, E. (2002), Placing search in context: the concept revisited, *ACM Transactions on Information Systems (TOIS)*, Volume 20, Issue 1, January 2002; Pages 116-131.
- Frakes, W. B. (1992), Stemming Algorithms, *In Information Retrieval: Data Structures & Algorithms* edited by William B. Frakes and Ricardo Baeza-Yates, p.504, Prentice Hall.
- Freihat, A. A., Abbas, M., Bella, G. and Giunchiglia, F. (2018), Towards an Optimal Solution to Lemmatization in Arabic, *Procedia Computer Science*, Vol. 142, pp. 132–140.
- Gabrilovich E. and Markovitch, S. (2009), Wikipedia-based Semantic Interpretation for Natural Language Processing, *Journal of Artificial Intelligence Research*, Vol. 34, pp. 443-498.
- Gabrilovich, E., & Markovitch, S. (2007). Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis, In: M.M. Veloso (ed.) *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 1606–1611 (2007)
- Goksel-Canbek, N., and Mutlu, M. E. (2016). On the track of Artificial Intelligence: Learning with Intelligent Personal Assistants, *International Journal of Human Sciences*, Volume 13(1), pp. 592-601. <https://doi:10.14687/ijhs.v13i1.3549>
- Gomaa, W. H., & Fahmy, A. A. (2013), A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, Vol. 68(13), pp.13-18.
- Guha, S., Rastogi, R. and Shim, K.(1998), CURE: An Efficient Clustering Algorithm for Large Databases, *In Proceedings of 1998 ACM-SIGMOD International Conference on Management of Data*, 1998.
- Guha, S., Rastogi, R. and Shim, K.(1999), ROCK: a robust clustering algorithm for categorical attributes, *In Proceedings of the 15th International Conference on Data Engineering*, 1999.

- Gupta, A., Kumar, A., & Gautam, J. (2017), A Survey on Semantic Similarity Measures, *International Journal for Innovative Research in Science & Technology*, Vol. 3(12), pp.243-247.
- Gupta, D., Kumar, R., Yadav, R. and Sajan, N. (2012), “Improving Unsupervised Stemming by using Partial Lemmatization Coupled with Data-based Heuristics for Hindi,” *International Journal of Computer Applications*, vol. 38, pp. 1-8, 2012.
- Hahn, U. and Heit, E. (2015), Cognitive Psychology of Semantic Similarity, *International Encyclopedia of the Social & Behavioral Sciences* (Second Edition), pp.579-584. <https://doi.org/10.1016/B978-0-08-097086-8.53026-8>
- Han, J., Kamber M. and Pei, J. (2012), Getting to Know Your Data. In *Data Mining: Concepts and Techniques*, The Morgan Kaufmann Series in Data Management Systems, pp.39-82.
- Harispe S., Ranwez, S., Janaqi, S. and Montmain,J. (2015), “Semantic similarity from natural language and ontology analysis”, *Synthesis Lectures on Human Language Technologies*, Vol. 8, No. 1, (2015): pp.1-254. <http://doi.org/10.2200/S00639ED1V01Y201504HLT027>.
- Harispe S.; Ranwez S. Janaqi S.; Montmain J. (2015). Semantic Similarity from Natural Language and Ontology Analysis, *Synthesis Lectures on Human Language Technologies*. Vol 8(1), pp.1–254.
- Harzing, A. (2016) Microsoft Academic (Search): A Phoenix arisen from the ashes? *Scientometrics*, Vol. 108 (3), pp. 1637-1647.
- Henry, S., McQuilkin, A. and McInnes, B. T. (2019), Association measures for estimating semantic similarity and relatedness between biomedical concepts, *Artificial Intelligence in Medicine*, pp.1-10, <https://doi.org/10.1016/j.artmed.2018.08.006>
- <https://doi.org/10.1145/3233547.3233640>
- Hug, S. E., Ochsner, M. and Braendle, M. P. (2016 )Citation Analysis with Microsoft Academic." arXiv Preprint arXiv:1609.05354. <https://arxiv.org/abs/1609.05354>
- Ingason A. K., Johannsson S. B., Rognvaldsson, E., Helgadóttir, S. and Loftsson H. (2009), Context-sensitive spelling correction and rich morphology. *Proceedings of NODALIDA*, Vol. 17.
- Ingason, A. K., Helgadóttir, S., Loftsson, H. and Rognvaldsson, E. (2008), A Mixed Method Lemmatization Algorithm Using a Hierarchy of Linguistic Identities (HOLI), *Advances in Natural Language Processing*, pp.205–216.
- Inkpen, D.(2007), Semantic Similarity Knowledge and its Applications. *Studia Univ. Babeş Bolyai, Informatica*, Volume LII, Number 1, 2007, pp. 11–22.
- Jiang, Y., Bai, W., Zhang, X. and Hu, J. (2017), Wikipedia-based information content and semantic similarity computation, *Information Processing and Management*, Vol. 53, pp. 248–265, <http://dx.doi.org/10.1016/j.ipm.2016.09.001>
- Johnson, S. C., (1967), Hierarchical clustering schemes, *Psychometrika*, September 1967, Volume 32, Issue 3, pp. 241–254. <https://doi.org/10.1007/BF02289588>
- Juckett, D. (2012), A method for determining the number of documents needed for a gold standard corpus, *Journal of Biomedical Informatics*, Volume 45, pp.460-470, doi:10.1016/j.jbi.2011.12.010.
- Karypis, G., Han E. and Kumar, V. (1999) Chameleon: hierarchical clustering using dynamic modeling, *Computer*, Volume 32, Issue 8, pp. 68 – 75. <https://doi.org/10.1109/2.781637>
- Kim, J. and Storey, V. C. (2011), Construction of Domain Ontologies: Sourcing the World Wide Web, *International Journal of Intelligent Information Technologies*, Volume 7(2), pp. 1-24.

- Kousha, K., Thelwall, M. & Abdoli, M. (2018), Can Microsoft Academic assess the early citation impact of in-press articles? A multi-discipline exploratory analysis, *Journal of Informetrics*, Vol. 12(1), pp.287-298.
- Kousha, K., & Thelwall, M. (2018), Can Microsoft Academic help to assess the citation impact of academic books? *Journal of Informetrics*, Vo.12(3), pp.972-984.
- Lee, C., Chang, J. W. and Hsieh, T. C. (2014), A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences Ming, *The Scientific World Journal*, Volume 2014, Article ID 437162, 17 pages <http://dx.doi.org/10.1155/2014/437162>.
- Lee, L. (1999), Measures of distributional similarity, *In Proceedings of the Annual Meeting of the Association for Computational Linguistics* (pp. 25–32). Morristown, NJ: Association for Computational Linguistics.
- Lemke, J. (1995), Intertextuality and text semantics, *In M. Gregory and P. Fries, Eds. Discourse in Society: Functional Perspectives*, Norwood, NJ: Ablex Publishing. 1995, pp.85-114.
- Leydesdorff, L. (2008), On the Normalization and Visualization of Author Co-Citation Data: Salton's Cosine versus the Jaccard Index, *Journal of The American Society for Information Science and Technology*, Vol. 59 (1), pp.77–85.
- Li, B. and Han, L. (2013). Distance weighted cosine similarity measure for text classification. Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B., & Yao, X., editors, *IDEAL*, volume 8206 of *Lecture Notes in Computer Science*, Springer, pp.611-618.
- Liddy, E.D. (2001), Natural Language Processing, in *Encyclopedia of Library and Information Science*, 2nd Ed. NY. Marcel Decker, Inc.
- Likas, A., Vlassis, N. and Verbeek, J. (2001), The global k-means clustering algorithm, [Technical Report] IAS-UVA-01-02, 2001, pp.12. [ffinria-00321515f](http://ffinria-00321515f).
- Lovins, J. (1968), Development of a Stemming Algorithm, *Mechanical Translation and Computational Linguistics*, Vol.11, pp. 22-31.
- Lovins, J. B. (1968), Development of a Stemming Algorithm, *Mechanical Translation and Computational Linguistics*, Vol.11, Nos.1 & 2, pp. 22- 31, March-June 1968.
- Lu, H., Li, Y., Chen, M., Kim, H. and Serikawa, S. (2018), Brain Intelligence: Go beyond Artificial Intelligence, *Mobile Networks and Applications*, Volume 23, Issue 2, pp 368–375; <https://doi.org/10.1007/s11036-017-0932-8>.
- Majumder, G., P. Pakray, A. Gelbukh and D. Pinto (2016), Semantic Textual Similarity Methods, Tools, and Applications: A Survey, *Computación y Sistemas*, Vol. 20(4), pp. 647–665. <https://doi.org/10.13053/CyS-20-4-2506>
- Malaviya, C., Shijie, W. and Cotterell, R. (2019), A Simple Joint Model for Improved Contextual Neural Lemmatization. [arXiv:1904.02306v2](https://arxiv.org/abs/1904.02306v2) [cs.CL]. (accessed on 29 July 2019).
- Manjavacas, E., Kadar, A. and Kestemont, M. (2019), Improving Lemmatization of Non-Standard Languages with Joint Learning, *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long Papers), Association for Computational Linguistics.
- Manning, C. D., Raghavan, P. and Schütze, (2009), An Introduction to information retrieval, Cambridge University Press Cambridge, England, Online edition (c) 2009.
- Manning, C. D., Raghavan, P. and Schütze, H. (2009), Hierarchical Clustering (Chapter-17) in *Introduction to Information Retrieval*, Cambridge University Press, pp.371-401.
- Martinez-Gil, J. (2014), An overview of textual semantic similarity measures based on web intelligence, *Artificial Intelligence Review*, Vol. 42(4), pp.935–943.

- Maynard D. G. and Ananiadou S. (1999), A linguistic approach to context clustering. *In Proceedings of Natural Language Processing Pacific Rim Symposium (NLPRS)*, pp. 346-351, Beijing, China.
- Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI'06*, pp. 775-780.
- Montes y Gómez, M., Gelbukh, A., López López, A and Baeza-Yates, R. (2001), Flexible Comparison of Conceptual Graphs. In Mayr, H.C., Lazansky, J., Quirchmayr, G., Vogel, P. (Eds.), *Proc. DEXA-2001, 12th International Conference and Workshop on Database and Expert Systems Applications. Lecture Notes in Computer Science, N 2113*, Springer-Verlag, 2001, pp. 102-111.
- Mottukuri, R., Nagaraju. M and Chilukuri, M. (2016), Similarity Measure for Text Classification, *International Journal of Emerging Trends & Technology in Computer Science*, Volume 5, Issue 6, pp.16-24.
- Naik, M. P., Prajapati, H. B. and Dabhi, V. K. (2015), A survey on semantic document clustering, *In Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference*, pp. 1-10. IEEE, 2015.
- Nalawade, R., Samal, A. and Avhad, K. (2016), Improved Similarity Measure for Text Classification And Clustering, *International Research Journal of Engineering and Technology*, Volume 3(5), pp.214-219.
- Niwattanakul, S., Singthongchai, J., Naenudorn, E. and Wanapu, S. (2013), Using of Jaccard Coefficient for Keywords Similarity, *Proceedings of the International MultiConference of Engineers and Computer Scientists 2013*, Vol. I, IMECS 2013, March 13 - 15, 2013, Hong Kong.
- Orduña Malea, E., Martín-Martín, A., Ayllón, J. M. and Delgado-López-Cózar, E. (2014). The silent fading of an academic search engine: the case of Microsoft Academic Search, *Online Information Review*. Vol 38(7), pp. 936-953. <https://doi:10.1108/OIR-07-2014-0169>
- Orkphol, K. and Yang, W. (2019), Word Sense Disambiguation Using Cosine Similarity Collaborates with Word2vec and WordNet, *Future Internet* 2019, 11(5), 114; <https://doi.org/10.3390/fi11050114>.
- Ortega, J. L. (2014), Influence of co-authorship networks in the research impact: Ego network analyses from Microsoft Academic Search, *Journal of Informetrics*, Vol. 8(3), pp. 728-737.
- Ortega, J. L. and Aguillo, I. F. (2014), Microsoft Academic Search and Google Scholar Citations: Comparative Analysis of Author Profiles. *Journal of the Association for Information Science & Technology*, Vol. 65 (6), pp.1149-1156. <https://doi.org/10.1002/asi.23036>.
- Pandey, P. and Singh, I. (2016), Comparison between Standard K-Mean Clustering and Improved K-Mean Clustering, *International Journal of Computer Applications*, Volume 146, No.13, pp.39-42.
- Paroubek, P., Chaudiron, S. and Hirschman, L. (2007), Principles of Evaluation in Natural Language Processing. *Traitement Automatique des Langues, ATALA*, 2007, 48 (1), pp.7-31.
- Pedersen, T. Pakhomov, S., Patwardhan, S. and Chuteb, C. G. (2007), Measures of semantic similarity and relatedness in the biomedical domain, *Journal of Biomedical Informatics*, Volume 40, Issue 3, June 2007, Pages 288-299. <https://doi.org/10.1016/j.jbi.2006.06.004>
- Periñán-Pascual, C and Arcas-Túnez, F. (2007), Deep semantics in an NLP knowledge base, *In Proceedings of the 12th Conference of the Spanish Association for Artificial Intelligence*, Salamanca, pp. 279--288.



- Periñan-Pascual, C. and Arcas-Túnez, F. (2010), The Architecture of FunGram KB. *In Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pp. 2667-2674. Valetta, European Language Resources Association.
- Pilehvar, M.T., and Navigli, R. (2015). An open-source framework for multilevel semantic similarity measurement. In *Proceedings of NAACL-HLT*, Denver, Colorado, pp. 76–80. © Association for Computational Linguistics.
- Pilehvar, M.T., Jurgens, D. and Navigli, R. (2013). Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *ACL (1)*, Sofia, Bulgaria, pp.1341–1351. © Association for Computational Linguistics.
- Powers, D. M. (2015), What the F-measure Doesn't Measure: Features, Flaws, Fallacies And Fixes, arXiv:1503.06410.
- R. M. Esteves, T. Hacker, and C. Rong, (2013), “Competitive k-means, a new accurate and distributed k-means algorithm for large datasets,” *In 2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, Vol. 1, Dec 2013, pp. 17–24.
- Roth, R. R., Lu, L., Farag, A., Sohn A. and Summers R. M. (2016) Spatial Aggregation of Holistically-Nested Networks for Automated Pancreas Segmentation, urn:arXiv:1606.07830.
- Roux M. (2015), A comparative study of divisive hierarchical clustering algorithms, Ithaca: Cornell University Library; 2015.
- Rus, V. et al. (2013). SEMILAR: The semantic similarity toolkit, *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 163–168, Sofia, Bulgaria, August 4-9, 2013, ©Association for Computational Linguistics.
- Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W. and Andruszkiewicz, P. (2016), Samsung Poland NLP Team at SemEval-2016 Task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity, *In Proceedings of SemEval-2016*. <http://www.aclweb.org/anthology/S16-1091>
- Saiyad, N. Y., Prajapati, H. B. and Dabhi, V. K. (2016), A survey of document clustering using semantic approach, *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, 2016, pp. 2555-2562.
- Sasaki, Y. (2007), The truth of the F-measure, Technical report, University of Manchester, School of Computer Science.
- Scheidsteger, T., Haunschild, R., Hug, S., & Bornmann, L. (2018). The concordance of fieldnormalized scores based on web of science and microsoft academic data: A case study in computer sciences. Paper presented at the International Conference on Science and Technology Indicators (STI 2018), Leiden, The Netherlands. <https://openaccess.leidenuniv.nl/handle/1887/65358>
- Sebastiani, F. (2002), Machine learning in automated text categorization, *ACM Computing Surveys*, Vol. 34 (1), pp.1-47.
- Seddah, D., Roux, J. L. and Sagot, B. (2012), Towards using data driven lemmatization for statistical constituent parsing of Italian, *In Working Notes of EVALITA 2012*, Rome, Italy, December 2012.
- Sharma, A. and López, Y. and Tatsuhiko Tsunoda, T. (2017), Divisive hierarchical maximum likelihood clustering, *BMC Bioinformatics* 2017, Vol. 18 (Supplement16):546, pp.14-147. <https://doi.org/10.1186/s12859-017-1965-5>.
- Sharma, R. and Srivastava, M. (2017), Testing the limits of unsupervised learning for semantic similarity, arXiv:1710.08246 [cs.CL].

- Sidorov, G., Gelbukh, E. and Pinto, D. (2014), Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model, *Computación y Sistemas*, Vol. 18, pp.491–504.
- Sil, A., Huang, F. and Yates, A. (2010), Extracting action and event semantics from web text, *In Proceedings of AAAI Fall Symposium on Commonsense Knowledge 2010*.
- Singhal, A. (2001), “Modern Information Retrieval: A Brief Overview”, *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, Vol. 24 (4), pp.35–43.
- Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B. and Wang, K. (2015), An Overview of Microsoft Academic Service (MAS) and Applications, *International World Wide Web Conference Committee (IW3C2)*. WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
- Sinoara, R., Antunes, J. and Rezende, S. (2017) Text mining and semantics: a systematic mapping study, *Journal of the Brazilian Computer Society*, Vol 23(9), <https://doi.org/10.1186/s13173-017-0058-7>
- Soğancıoğlu, G., Öztürk H. and Özgür, A. (2017), BIOSSES a semantic sentence similarity estimation system for the biomedical domain, *Bioinformatics*, Volume 33, Issue 14, 15 July 2017, Pages i49–i58, <https://doi.org/10.1093/bioinformatics/btx238>
- Sreepathi, S., Kumar, J., Mills, R. T., Hoffman, F. M., Sripathi, V., and Hargrove, W. W. (2017), *Parallel Multivariate Spatio-Temporal Clustering of Large Ecological Datasets on Hybrid Supercomputers*. IEEE Cluster 2017 - Honolulu, Hawaii, USA. doi:10.1109/CLUSTER.2017.88.
- Stavrianou, A., Andritsos, P. and Nicoloyannis, N. (2007) *Overview and semantic issues of text mining*, ACM SIGMOD, Volume 36, Issue 3, pp.23-34.
- T Müller, R Cotterell, A Fraser, H Schütze (2015), Joint Lemmatization and Morphological Tagging with Lemming, *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Lisbon, Portugal, pp. 2268–2274.
- Tahamtan, I. and Bornmann, L. (2019), What Do Citation Counts Measure? An Updated Review of Studies on Citations in Scientific Documents Published between 2006 and 2018, arXiv preprint arXiv:1906.04588.
- Takale, S. A. and Nandgaonkar, S. A. (2010) Measuring semantic similarity between words using web documents, *International Journal of Advanced Computer Science and Applications*, Vol. 1, No.4, pp.78-85.
- Thelwall, M. (2018), Microsoft Academic automatic document searches: accuracy for journal articles and suitability for citation analysis. *Journal of Informetrics*, Vol. 12(1), pp.1-7. doi:10.1016/j.joi.2017.11.001
- Thiagarajan, R., Manjunath, G., and Stumptner, M. (2008), Computing Semantic Similarity Using Ontologies *International Semantic Web Conference (ISWC)*, (pp. 2-16), Kalsruhe, Germany.
- Uschold, M. and Gruninger, M. (1996). *Ontologies: Principles, Methods and Applications*, *Knowledge Engineering Review*, Vol. 11(2), June 1996.
- Vairavasundaram, S. and Logesh R. (2018), Applying Semantic Relations for Automatic Topic Ontology Construction, *In Developments and Trends in Intelligent Technologies and Smart Systems*, doi:10.4018/978-1-5225-3686-4.ch004.
- van Rijsbergen, C. J. (1979), *Information Retrieval*, London: Butterworths, 1979. <http://www.dcs.gla.ac.uk/Keith/Preface.html> (retrieved on 25 November 2019).
- Ward, J., Bejarano, W. and Dudás, A. (2015), Scholarly Social Media Profiles and Libraries: A Review, *Liber Quarterly*, Vol. 24 (4), pp.174–204. <https://doi.org/10.18352/lq.9958>.

- Wei, T., Lu, Y., Chang, H., Zhoua, Q. and Bao, X. (2015) A semantic approach for text clustering using WordNet and lexical chains, *Expert Systems with Applications*, Volume 42, Issue 4, pp. 2264-2275.
- Willett, P. (2006), The Porter stemming algorithm: then and now, *Program: electronic library and information systems*, Vol. 40 (3). pp. 219-223.
- Winston, G. P.; Cardoso, M Jorge; Williams, Elaine J; Burdett, Jane L; Bartlett, Philippa A; Espak, Miklos; Behr, Charles; Duncan, John S; Ourselin, Sebastien (2013), Automated hippocampal segmentation in patients with epilepsy, *Epilepsia*, Vol. 54(12), pp.2166–2173.
- Wissler, L., Almashraee, M., Díaz, D.M., and Paschke, A.(20140, The gold standard in corpus annotation. In: Institute of Electrical and Electronics Engineers Germany Student Conference
- Xu, J. and Lu, Q. (2013), Second Joint Conference on Lexical and Computational Semantics (SEM), Volume 1: *Proceedings of the Main Conference and the Shared Task*, pp.90–95, Atlanta, Georgia, June 13-14. Association for Computational Linguistics.
- Yang, Y. and Tar, C. (2018), Advances in Semantic Textual Similarity, <https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html>, posted on 17 May 2018.
- Zhao, Y. and Karypis, G. (2002), Comparison of agglomerative and partitional document clustering algorithms, *Technical Report* 02-014, University of Minnesota.
- Zhou, X., Han, H., Chankai, I., Prestrud, A. and Brooks, A. (2006), Approaches to Text Mining for Clinical Medical Records, 21st Annual ACM Symposium on Applied Computing 2006, Technical tracks on Computer Applications in Health Care (CAHC 2006), Dijon, France, April 23 -27, 2006.
- Zou, K. H., Warfield, S.K, Bharatha, A., Tempany, C.M.C, Kaus, M.R., Haker, S.J., Wells, W. M., Jolesz, F.A. and Kikinis, R. (2004), Statistical validation of image segmentation quality based on a spatial overlap index, *Academic Radiology*, Vol. 11, No. 2, pp.178–189.

#### Reference Websites:

1. P. J. Hancox, A brief history of Natural Language Processing, [Online] [https://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1\\_history.html](https://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1_history.html) (accessed on 15 June 2019)
2. <https://medium.com/@datamonsters/artificial-neural-networks-in-natural-language-processing-bcf62aa9151a> (accessed on 15 June 2019).
3. <https://emerj.com/ai-sector-overviews/natural-language-processing-business-applications>(accessed on 18 June 2019)
4. <https://brotherfish.me/portfolio/lexical-semantics/>(accessed on 19 June 2019)
5. <https://www.expertsystem.com/examples-natural-language-processing-systems-artificial-intelligence/>(accessed on 20 June 2019)
6. <https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/home>(accessed on 22<sup>nd</sup> June 2019).
7. <https://www.microsoft.com/en-us/research/project/academic/articles/microsoft-academic-increases-power-semantic-search-adding-fields-study/> (accessed on 22<sup>nd</sup> June 2019).
8. Scikit-learn. classification.py.<https://github.com/scikit-learn/scikit-learn/blob/0.20.X/sklearn/metrics/classification.py> (accessed on 22<sup>nd</sup> September 2019)
9. Scikit-learn. Machine learning in Python. <https://scikit-learn.org> (accessed on 22<sup>nd</sup> September 2019)

10. Scikit-learn. Precision, recall and F-measures. [https://scikit-learn.org/stable/modules/model\\_evaluation.html#precision-recall-f-measure-metrics](https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics) (accessed on 22<sup>nd</sup> September 2019)



# Appendix

## Appendix A: Gold Standard Data

| List of sentences:   | Group Number |
|--|--------------|
| A parity friendly corporate culture  | 1            |
| Coordination by the planning team  | 1            |
| A presentation outlining the benefits of gender parity                           | 1            |
| Generation of a financial estimation of potential benefits                       | 1            |
| Focus on lowering the attrition rate   | 1            |
| Creating parity issues awareness for top management                              | 1            |
| Advertise the campaign   | 3            |
| Dedicated presentation of the gender parity benefits                             | 1            |
| Corporate support for parity   | 1            |
| Having a team member realizing a financial estimation of the benefits            | 3            |
| Top Management support to the new policy   | 1            |
| Offer flexible working arrangements  | 5            |
| A communication professional involved on the communication strategy and benefits | 3            |
| A lawyer stating the emerging compliance issues concerning gender parity         | 2            |
| Dedicated head-hunter team   | 4            |
| Exposing women to all company operations and functions                           | 6            |
| Identifying women to be involved in cross-functional projects                    | 6            |
| Making top management aware of the current gender parity issues                  | 1            |
| Legal analysis regarding compliance issues of gender parity                      | 2            |
| Create a recruitment team  | 4            |

| List of sentences:   | Group Number |
|--|--------------|
| Include new benefits for women   | 5            |
| Assigning women managers visible and challenging tasks                       | 6            |
| Involving women in specific cross functional projects (internal or external) | 6            |
| Hire a lawyer to analyze compliance issues with regards to gender parity     | 2            |
| Parity friendly corporate environment  | 1            |
| Create a head-hunting team   | 4            |
| Launch internal recruitment campaign   | 6            |
| Create a benefits package for female employees                               | 5            |
| Sufficient advertisement for the new campaign                                | 3            |
| Generate a list of internal female candidates                                | 6            |
| Analysis of current recruitment processes                                    | 4            |
| Provide better benefits for female employees                                 | 5            |
| Start an external recruitment campaign                                       | 4            |
| List current female employees that can be targeted for promotion             | 6            |

|   |   |
|---|---|
| Create report analyzing current recruitment processes                 | 4 |
| Lower attrition rate  | 5 |
| Legal analysis of potential compliance issues cornering gender parity | 2 |
| Identify teams/departments to recruit for                             | 4 |
| Building awareness of the gender parity issues to the management      | 1 |
| Run an analysis of the current recruitment process                    | 4 |

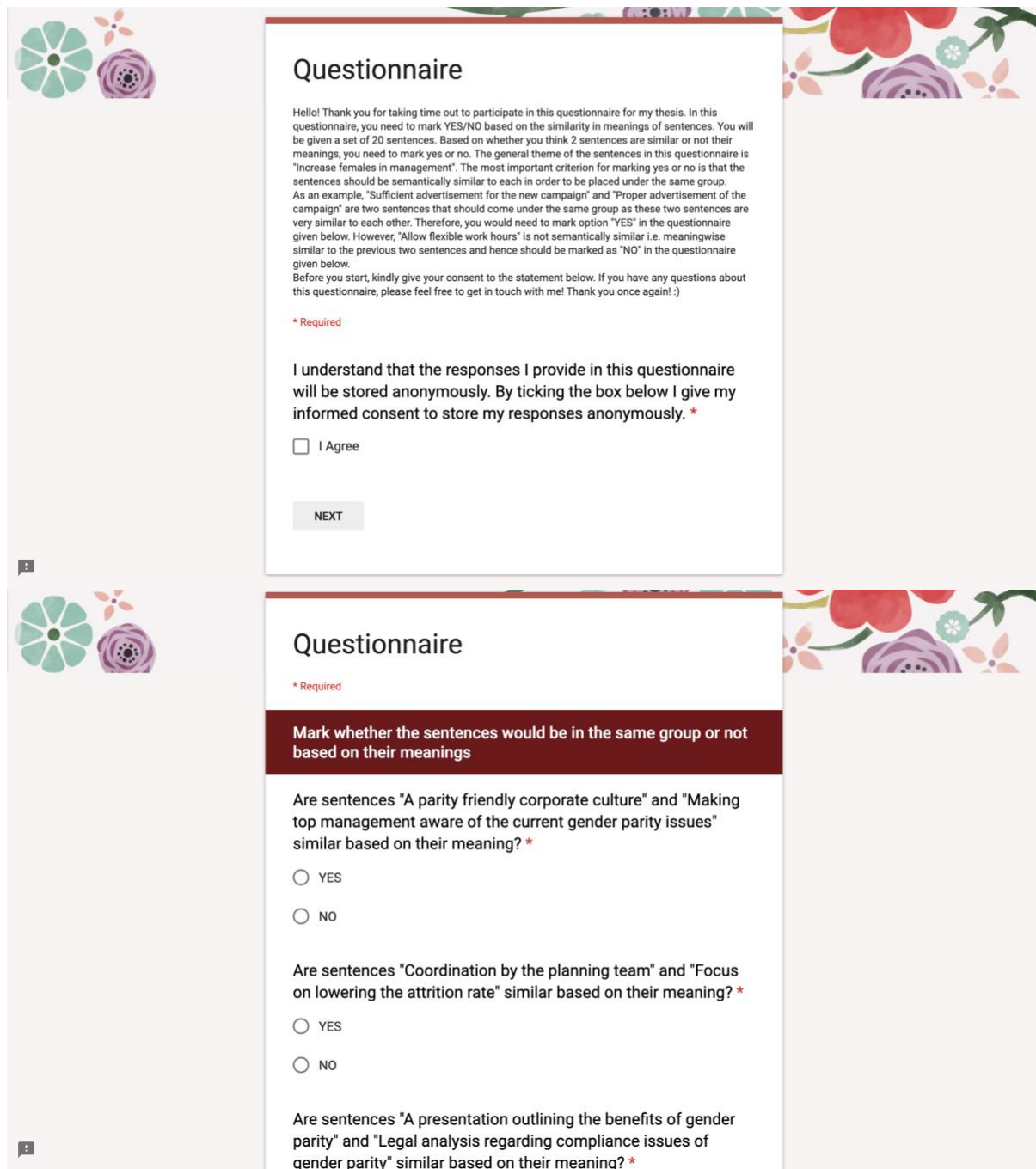
| <b>List of sentences:</b>  | <b>Group Number</b> |
|--|---------------------|
| Collaboration with middle management of each of the functional divisions | 1                   |
| List of female employees to be promoted established by HR department     | 7                   |
| Collaboration with middle management of the various departments          | 1                   |
| Put effort in lowering attrition rate                                    | 6                   |
| Dedicated HR team to process internal HR system information              | 4                   |
| External recruitment campaign  | 4                   |
| Lawyer analysis of the potential compliance issues of gender parity      | 2                   |
| Gaining support from the top management                                  | 1                   |
| Provide training workshops for current female employees                  | 7                   |
| Allow more flexible work arrangements                                    | 6                   |
| To rise management awareness over the gender parity issues               | 1                   |
| Significant pay parity   | 5                   |
| Trainings offered to broader female employees' skill set                 | 7                   |
| Organize informational events  | 3                   |
| Put together a dedicated recruitment team                                | 4                   |
| Raising of selected women's salaries                                     | 5                   |
| Improve current benefits for women employees                             | 6                   |
| Offer training to female employees                                       | 7                   |
| Higher salaries for positions showing large pay disparity for women      | 5                   |
| Available resources within the HR department                             | 6                   |

| <b>List of sentences:</b>  | <b>Group Number</b> |
|--|---------------------|
| Initiate training opportunities for female employees   | 5                   |
| Start a campaign to boost internal recruitment   | 5                   |
| Financial analysis of payroll parity measures  | 3                   |
| Better salaries for women in positions showing large disparity   | 3                   |
| Support of the policy from the top management  | 1                   |
| Higher salaries for selected female employees  | 3                   |
| Communicating new values regarding gender parity   | 2                   |
| Increasing women's salaries for positions showing disparity  | 3                   |
| Top-management involvement for gender equality values communicated by internal and external means of communication | 2                   |
| Decrease men's salaries for positions showing disparity  | 3                   |
| Top management appearance in various media   | 1                   |

|  |   |
|--|---|
| Promoting women from level 3 and 4                         | 5 |
| Event organization   | 2 |
| Dedicated content wrote for top managers, signed by them   | 1 |
| To establish the financial means allocated to this purpose | 1 |
| Dedicated video produced and diffused                      | 2 |
| Focus on promotions of women from levels 3 and 4           | 5 |
| Executive training for selected female employees           | 5 |
| Communication action plan and tactics                      | 2 |
| Allow for flexible work hours                              | 4 |

| <b>List of sentences:</b>  | <b>Group Number</b> |
|--|---------------------|
| Drawing a requirement list for female employees to fit the “promotable profile” before being trained | 5                   |
| Accountable HR team for the roll out   | 4                   |
| Recruiting more women  | 3                   |
| Defining an accountable manager  | 1                   |
| In-depth analysis: Distribution of women across functions  | 1                   |
| Creation of team tasked with the policy roll out   | 1                   |
| Focus on hiring more females   | 3                   |
| Creation of a dedicated HR team to carry out rolling out the policy                                  | 1                   |
| Recruit new female candidates  | 3                   |
| Launch external recruitment campaign   | 3                   |
| Specific internal recruitment process to design/apply  | 5                   |
| Recruit more female candidates   | 3                   |
| List of eligible female candidates   | 5                   |
| Focus hiring efforts on women  | 3                   |
| Proper advertisement of the campaign   | 2                   |
| Create a list of eligible female employees   | 5                   |
| External campaign to reach skilled candidates  | 3                   |
| Focus on recruiting female candidates  | 3                   |
| List of female employees to be targeted  | 5                   |
| Stating the current situation  | 1                   |

# Appendix B: Yes/No Questionnaire



The image shows two screenshots of a questionnaire form. The top screenshot is the introductory page, and the bottom screenshot is the first question page. Both pages feature a light pink background with floral illustrations in the corners. The top-left corner has a green citrus slice and a purple rose. The top-right corner has a red apple and a purple rose. The bottom-left corner has a green citrus slice and a purple rose. The bottom-right corner has a red apple and a purple rose.

## Questionnaire

Hello! Thank you for taking time out to participate in this questionnaire for my thesis. In this questionnaire, you need to mark YES/NO based on the similarity in meanings of sentences. You will be given a set of 20 sentences. Based on whether you think 2 sentences are similar or not their meanings, you need to mark yes or no. The general theme of the sentences in this questionnaire is "Increase females in management". The most important criterion for marking yes or no is that the sentences should be semantically similar to each in order to be placed under the same group. As an example, "Sufficient advertisement for the new campaign" and "Proper advertisement of the campaign" are two sentences that should come under the same group as these two sentences are very similar to each other. Therefore, you would need to mark option "YES" in the questionnaire given below. However, "Allow flexible work hours" is not semantically similar i.e. meaningwise similar to the previous two sentences and hence should be marked as "NO" in the questionnaire given below. Before you start, kindly give your consent to the statement below. If you have any questions about this questionnaire, please feel free to get in touch with me! Thank you once again! :)

**\* Required**

I understand that the responses I provide in this questionnaire will be stored anonymously. By ticking the box below I give my informed consent to store my responses anonymously. \*

I Agree

NEXT

---

## Questionnaire

**\* Required**

**Mark whether the sentences would be in the same group or not based on their meanings**

Are sentences "A parity friendly corporate culture" and "Making top management aware of the current gender parity issues" similar based on their meaning? \*

YES

NO

Are sentences "Coordination by the planning team" and "Focus on lowering the attrition rate" similar based on their meaning? \*

YES

NO

Are sentences "A presentation outlining the benefits of gender parity" and "Legal analysis regarding compliance issues of gender parity" similar based on their meaning? \*

Are sentences "A presentation outlining the benefits of gender parity" and "Legal analysis regarding compliance issues of gender parity" similar based on their meaning? \*

YES

NO

Are sentences "Having a team member realizing a financial estimation of the benefits" and "A communication professional involved on the communication strategy and benefits" similar based on their meaning? \*

YES

NO

Are sentences "Dedicated head-hunter team" and "Create a recruitment team" similar based on their meaning? \*

YES

NO

Are sentences "Offer flexible working arrangements" and "Corporate support for parity" similar based on their meaning? \*

YES

Are sentences "Offer flexible working arrangements" and "Corporate support for parity" similar based on their meaning? \*

YES

NO

Are sentences "Identifying women to be involved in cross-functional projects" and "Exposing women to all company operations and functions" similar based on their meaning? \*

YES

NO

Are sentences "Creating parity issues awareness for top management" and "Top Management support to the new policy" similar based on their meaning? \*

YES

NO

Are sentences "Advertise the campaign " and "Generation of a financial estimation of potential benefits" similar based on their meaning? \*

YES

Are sentences "Advertise the campaign " and "Generation of a financial estimation of potential benefits" similar based on their meaning? \*

YES

NO

Are sentences "Dedicated presentation of the gender parity benefits" and "A lawyer stating the emerging compliance issues concerning gender parity" similar based on their meaning? \*

YES

NO

Are sentences "Making top management aware of the current gender parity issues" and "Identifying women to be involved in cross-functional projects" similar based on their meaning? \*

YES

NO

Are sentences "Generation of a financial estimation of potential benefits" and "Having a team member realizing a financial estimation of the benefits " similar based on their meaning? \*

Are sentences "Generation of a financial estimation of potential benefits" and "Having a team member realizing a financial estimation of the benefits " similar based on their meaning? \*

YES

NO

Are sentences "Dedicated presentation of the gender parity benefits" and "A presentation outlining the benefits of gender parity" similar based on their meaning? \*

YES

NO

Are sentences "Legal analysis regarding compliance issues of gender parity" and "A lawyer stating the emerging compliance issues concerning gender parity " similar based on their meaning? \*

YES

NO

Are sentences "Top Management support to the new policy" and "Exposing women to all company operations and functions" similar based on their meaning? \*

Are sentences "Top Management support to the new policy" and "Exposing women to all company operations and functions" similar based on their meaning? \*

YES

NO

Are sentences "Focus on lowering the attrition rate" and "Advertise the campaign" similar based on their meaning? \*

YES

NO

Are sentences "Offer flexible working arrangements" and "Identifying women to be involved in cross-functional projects" similar based on their meaning? \*

YES

NO

Are sentences "A communication professional involved on the communication strategy and benefits" and "Having a team member realizing a financial estimation of the benefits" similar based on their meaning? \*

Are sentences "A communication professional involved on the communication strategy and benefits" and "Having a team member realizing a financial estimation of the benefits" similar based on their meaning? \*

YES

NO

Are sentences "A parity friendly corporate culture" and "Coordination by the planning team" similar based on their meaning? \*

YES

NO

Are sentences "Generation of a financial estimation of potential benefits" and "Having a team member realizing a financial estimation of the benefits" similar based on their meaning? \*

YES

NO

BACK

SUBMIT

# Appendix C: Individual Clustering Questionnaire

## Text Grouping Questionnaire

Hello! Thank you for taking time out to participate in this questionnaire for my thesis. In this questionnaire, you need to group texts based on the similarity in their themes. You will be given a set of 20 different sentences. Based on what you think are similar sentences, you need to group them in multiple groups. There are no restrictions on the number of groups that you should create but all sentences under a group should be related to the same topic. The general theme of the sentences in this questionnaire is "Increase females in management". However, there can be various subgroups that can be made under the general theme based on how similar two sentences are. You need not give a name to each group created but must make sure that the sentences under each group are similar to each other. Also, an important criterion for the grouping is that the sentences should be semantically similar to each in order to be placed under the same group. If you find that a sentence does not belong to any group, you are free to leave it as a separate one. Please do NOT forcefully add a sentence that you find semantically dissimilar to a group. As an example, "Sufficient advertisement for the new campaign" and "Proper advertisement of the campaign" are two sentences that should come under the same group as these two sentences are very similar to each other. Both of them would come under the group category "increasing females in management by carrying out proper advertisement for the campaign". Before you start, kindly give your consent to the statement below. If you have any questions about this questionnaire, please feel free to get in touch with me! Thank you once again! :)

**\* Required**

I understand that the responses I provide in this questionnaire will be stored anonymously. By ticking the box below I give my informed consent to store my responses anonymously. \*

I agree

**NEXT**

Never submit passwords through Google Forms.

## Text Grouping Questionnaire

**\* Required**

**Group the sentences based on their semantic similarity.**

List of sentences:

1. A parity friendly corporate culture
2. Coordination by the planning team
3. A presentation outlining the benefits of gender parity
4. Generation of a financial estimation of potential benefits
5. Focus on lowering the attrition rate
6. Creating parity issues awareness for top management
7. Advertise the campaign
8. Dedicated presentation of the gender parity benefits
9. Corporate support for parity
10. Having a team member realizing a financial estimation of the benefits
11. Top Management support to the new policy
12. Offer flexible working arrangements
13. A communication professional involved on the communication strategy and benefits
14. A lawyer stating the emerging compliance issues concerning gender parity
15. Dedicated head-hunter team
16. Exposing women to all company operations and functions
17. Identifying women to be involved in cross-functional projects
18. Making top management aware of the current gender parity issues
19. Legal analysis regarding compliance issues of gender parity
20. Create a recruitment team

Please provide a group heading for each new group created in the text box below. For example, Group 1, Group 2, etc. \*

Your answer

**BACK** **SUBMIT**

Never submit passwords through Google Forms.

This form was created inside of Universiteit Utrecht Studenten. [Report Abuse](#)

Google Forms



## Appendix D: Sentence Agreement Program

```
import csv
import pandas as pd
from sklearn.metrics import precision_recall_fscore_support

r_pred = pd.read_csv('/Users/applemacbook/Desktop/QA1.csv')
r_true = pd.read_csv('/Users/applemacbook/Desktop/GS1.csv')

def classes_true(value):
    classes={}
    for s in range(len(r_true['List of sentences:'])):
        sent=(r_true['List of sentences:'][s].strip())
        classes[(sent[sent.index('.')+2:])] = r_true['Group
Number'][s]
    return classes

def classes_pred(value):
    classes={}
    for t in range(len(r_pred['List of sentences:'])):
        sent=(r_pred['List of sentences:'][t].strip())
        classes[(sent[sent.index('.')+2:])] = r_pred['Group
Number'][t]
    return classes

y_true = classes_true(r_true)
y_pred = classes_pred(r_pred)

precision_recall_fscore_support(y_true, y_pred,
average='weighted')
```

For calculation of agreement between gold standard and yes/no questionnaire, only the last line of the code was changed to as below:

```
precision_recall_fscore_support(y_true, y_pred,
average='binary')
```