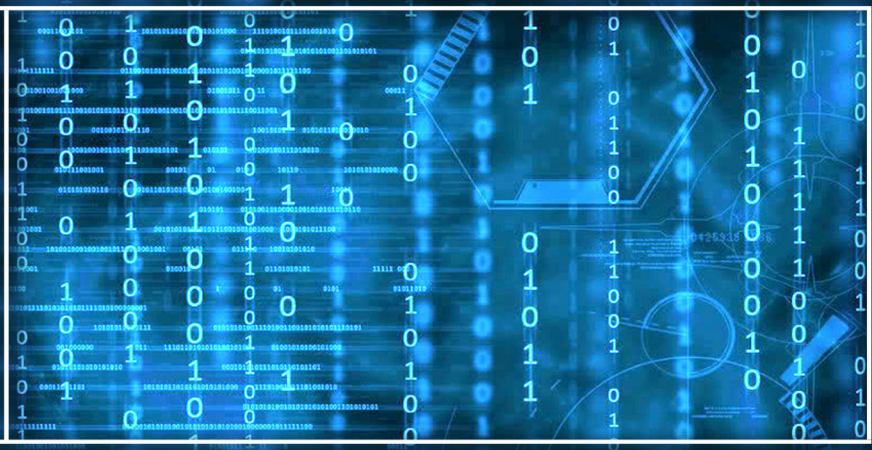
IJACSA
WHERE WISDOM SHARES
International Journal of Advanced Computer Science and Applications

Special Issue



Special Issue on Natural Language Processing 2014

ISSN 2156-5570(Online)
ISSN 2158-107X(Print)

SAI
www.ijacsa.thesai.org

IJACSA

WHERE WISDOM SHARES

INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS

THE SCIENCE AND INFORMATION ORGANIZATION

www.thesai.org | info@thesai.org

OAIster getCITED Google Scholar BETA BASE Bielefeld Academic Search Engine ULRICHSWEB GLOBAL SERIALS DIRECTORY arXiv.org

DOAJ DIRECTORY OF OPEN ACCESS JOURNALS IET InspecDirect INDEX COPERNICUS INTERNATIONAL WorldCat Window to the world's libraries Microsoft Academic Search EBSCO HOST Research Databases

Editorial Preface

From the Desk of Guest Editor …

NLP has got many new challenging and interesting areas of research. It has gone much beyond ordinary translation of text from one language to another. Existing translation tools are becoming more and more accurate and are preserving the context nicely. Till few years, there were only applications laced with other areas of AI like image processing (for OCR), speech recognition, etc. These days, due to the evolution of web technologies, there are many applications coming up using NLP as a key domain. Name entity recognition (NER) is a huge addition these days in almost languages.

In this Special Issue on Natural Language Processing, we have papers in the area of pure NLP as well as with application areas of soft computing, ontology, etc. It is good to see papers coming from all across the globe; they are from India, Bangladesh, Spain, Nepal, Brazil, UK and Belgium.

There is a paper by Sachin et al that uses Polar Fuzzy Neutrosophic Semantic Net (PFNSN) to represent a given sentence. Then the paper by Kameswara Rao et al. explains elaborately on the issues in splitting Telugu bi-gram words on the basis of vowels. Telugu is an ancient language widely spoken in southern India by over 80 million population. Mirdha et al in their paper described about usage of universal network language (UNL) to solve the semantics of a phrase. A very nice method of analyzing opinions and arguments in print media, which can be extended to other areas, was brought out by Bal Krishna. Robert in his paper on automating the process of shaping meta data is quite an interesting read and a work that can be used by all researchers attempting to discover words or shape meta data. There is a paper on use of NLP for building chemical textile ontology, very interesting work... Ivo et al described excellently on building new ontologies and the visualization using MDS were outstanding. I am confident that this work will bring many AI approaches together. Using hybrid representation of given story and then querying with questions to get answers were carried out Poonam et al. Finally, Matthew presented a very good exploration of simplifying text mining.

I thank the Managing Editor for inviting me to be Guest Editor for this Special Issue on Natural Language Processing.

We had an acceptance rate of 66% for the Special Issue. We will like to bring out such domain specific Special Issues to bring together findings and opinions of efforts made in similar research areas. We deeply appreciate the efforts done by many authors, those whose papers found place in this Journal as well as those who could not find.

My hearty greetings to all authors, reviewers, editors and the Journal staff for such an accomplishment.

Happy reading!!!

Prof. T. V. Prasad, PhD
Former Dean (Computing Sciences)
Visvodaya Technical Academy, India

IJACSA
Special Issue on Natural Language Processing 2014
ISSN 2156-5570 (Online)
ISSN 2158-107X (Print)

Editorial Board

CONTENTS

Representation of a Sentence using a Polar Fuzzy Neutrosophic Semantic Net

Sachin Lakra
Research Scholar
Computer Science & Engineering
K. L. University Vaddeswaram,
Guntur, AP, India

T. V. Prasad
Former Dean of Computing
Sciences, Visvodaya Technical
Academy, Kavali, AP,
India.

G. Ramakrishna
Professor Computer Science &
Engineering K. L. University
Vaddeswaram, Guntur, AP,
India

***Abstract*—A semantic net can be used to represent a sentence. A sentence in a language contains semantics which are polar in nature, that is, semantics which are positive, neutral and negative. Neutrosophy is a relatively new field of science which can be used to mathematically represent triads of concepts. These triads include truth, indeterminacy and falsehood, and so also positivity, neutrality and negativity. Thus a conventional semantic net has been extended in this paper using neutrosophy into a Polar Fuzzy Neutrosophic Semantic Net. A Polar Fuzzy Neutrosophic Semantic Net has been implemented in MATLAB and has been used to illustrate a polar sentence in English language. The paper demonstrates a method for the representation of polarity in a computer's memory. Thus, polar concepts can be applied to imbibe a machine such as a robot, with emotions, making machine emotion representation possible.**

Keywords—semantic net; polarity; neutrosophy; polar fuzzy neutrosophic semantic net; NLP

I. INTRODUCTION

Representation of polarity, that is, positivity, neutrality and negativity, of a sentence in a natural language, has been a long-standing problem in Natural Language Processing. Knowledge representation using various techniques including frames, conceptual dependency and semantic nets [1] were proposed 7-8 decades ago since the advent of AI. Numerous accounts of artificially intelligent machines which are incapable of emotion representation exist in the form of science fiction literature. Machines are considered to be incapable of possessing emotions in the same way as human beings.

Semantic nets were first proposed by Charles S. Peirce in the year 1909 [2]. Semantic nets were first invented for computers by Richard H. Richens of the Cambridge Language Research Unit in 1956 [3]. An extension of semantic nets to include inexactitude and imprecision was made by the development of Fuzzy Semantic Nets [4]. Fuzzy Cognitive Maps and Neutrosophic Cognitive Maps were introduced in 2003 by Kandasamy and Smarandache [5]. These maps introduced the notion of causality in a network structure to represent concepts and their interdependence. However, none of these attempts were able to incorporate polarity.

This paper extends traditional semantic nets into polar fuzzy neutrosophic semantic nets (PFNSN). The term "polarity" has been introduced to distinguish the triad of the concepts of positivity, neutrality and negativity from the triad of concepts of truth, indeterminacy and falsehood. Earlier, both these triads were categorized under the term neutrosophy. A PFNSN has been implemented in MATLAB to represent an English sentence which specifically depicts polar concepts. Applications of a PFNSN can be found in machine emotion representation and intelligent response generation.

II. NEUTROSOPHY

The name neutrosophy is derived from Latin "neuter" meaning neutral and Greek "sophia" meaning skill/wisdom. Neutrosophy is a branch of philosophy, introduced by Florentin Smarandache in 1980, which studies the origin, nature, and scope of neutralities, as well as their interactions with different ideational spectra [6].

Florentin Smarandache had generalized the fuzzy logic, and introduced two new concepts [7]:

- "neutrosophy" – study of neutralities as an extension of dialectics;
- and its derivative "neutrosophic", such as "neutrosophic logic", "neutrosophic set", "neutrosophic probability", and "neutrosophic statistics" and thus opened new ways of research in four fields: philosophy, logics, set theory, and probability/statistics.

Neutrosophy considers a proposition, theory, event, concept, or entity, "A" in relation to its opposite, "Anti-A" (and that which is also not A, "Non-A"), and that which is neither "A" nor "Anti-A", denoted by "Neut-A".

Neutrosophy is the basis of neutrosophic logic, neutrosophic probability, neutrosophic sets, and neutrosophic statistics.

Definition 1: Main Principle of Neutrosophy [6]

Between an idea <A> and its opposite <Anti-A>, there is a continuum-power spectrum of neutralities <Neut-A>.

Definition 2: Fundamental Thesis of Neutrosophy [6]

Any idea <A> is *t*% true, *i*% indeterminate, and *f*% false, where $t, i, f \in]^{-}0, 1^{+}[$.

Here, $^{-}0 = 0 - \varepsilon$ and $1^{+} = 1 + \varepsilon$, where ε is an infinitesimal value.Definition 3:Neutrosophic Components [10].

Let *T, I, F* be standard or non-standard real subsets of $]^{-}0, 1^{+}[$, where the sets *T, I, F* are not necessarily intervals, but

may be any real sub-unitary subsets: discrete or continuous; single-element, finite, or (countable or uncountable) infinite; union or intersection of various subsets; etc.

T, I, F, called neutrosophic components, represent the truth value, indeterminacy value, and falsehood value, respectively, referring to neutrosophy, neutrosophic logic, neutrosophic set, neutrosophic probability and neutrosophic statistics[10].

This representation is closer to the reasoning performed by the human mind. It characterizes/catches the imprecision of knowledge or linguistic inexactitude perceived by various observers (reason why *T, I, F* are subsets - not necessarily single-elements), uncertainty due to incomplete knowledge or acquisition errors or stochasticity (reason why the subset *I* exists), and vagueness due to lack of clear contours or limits (reason why *T, I, F* are subsets and *I* exists) [10].

One has to specify the superior (x_sup) and inferior (x_inf) limits of the subsets because in many problems the necessity to compute them arises.

III. MATHEMATICAL DEFINITION OF A POLAR FUZZY NEUTROSOPHIC SEMANTIC NET

A semantic net has a graph as its core mathematical structure. Consequently, since a neutrosophic semantic net is an extension of a conventional semantic net, its core mathematical structure is a neutrosophic graph.

A. Neutrosophic Graph

Definition 4: Graph

A graph is an ordered pair $G = (V, E)$ comprising set V of vertices or nodes together with a set E of edges or lines, which are 2-element subsets of V (i.e., an edge is related with two vertices, and the relation is represented as an unordered pair of the vertices with respect to the particular edge) [8, 9].

Definition 5: Neutrosophic Set [10]

A Neutrosophic Set is a set such that an element belongs to the set with a neutrosophic probability, i.e. *t*% is true that the element is in the set, *f*% false, and *i*% indeterminate.

Definition 6: Neutrosophic Indeterminacy

Indeterminacy is defined as the state of being defined in an inexact manner. Indeterminacy is represented in neutrosophy by a set I of values referring to the degree of inexactitude and is termed as Neutrosophic Indeterminacy.

Definition 7: Neutrosophic Point Graph [11]

A neutrosophic point graph G_N is a graph G with finite non empty set $V_N = V_N(G)$ of p-points where at least one of the points in $V_N(G)$ is an indeterminate node, element, point or vertex.

Note here that $V_N(G) = V(G) + N$, where $V(G)$ are points or vertices of the graph G and N the non-empty set of points which are indeterminate nodes.

Definition 8: Neutrosophic Edge Graph [11]

Let $V(G)$ be the set of all vertices of the graph G. If the edge set is $E(G)$, where at least one of the edges of G is an indeterminate one, then we call such graphs as neutrosophic edge graphs.

Thus, the neutrosophic vertex graph is distinctly different from the neutrosophic edge graph. They differ from each other on the edge set and the vertex set. The edge set of a neutrosophic vertex graph are all usual edges whereas only the vertex sets are indeterminate. On the contrary, the vertex set of the neutrosophic edge graph has the vertex set to be the usual set. The difference lies only in the edge set, where some of the edges are indeterminate [11]. A neutrosophic edge graph is simply referred to as a neutrosophic graph.

Definition 9: Neutrosophic Graph [11]

A neutrosophic graph is a graph in which at least one edge is an indeterminacy denoted by dotted lines.

Definition 10: Neutrosophic Directed Graph [11]

A neutrosophic directed graph is a directed graph which has at least one edge to be an indeterminacy.

Definition 11: Doubly or Strongly Neutrosophic Graph [11]

A graph G is said to be a doubly or strongly neutrosophic graph if the graph has both indeterminate vertices and indeterminate edges. The indeterminate edges are denoted by dotted lines whereas the indeterminate vertices are denoted by $N_1, \ldots, N_k$.

Definition 12: Neutrosophic Vertex Graph [11]

A neutrosophic vertex graph G_N is said to be neutrosophic simple if the graph has no loops and no multiple edges connecting an indeterminate vertex or two indeterminate vertices.

Definition 13: Neutrosophic Vertex [11]

Elements of $V_N(G)$, where $V_N(G)$ is a neutrosophic point graph, are called the neutrosophic vertices of G. The number of elements in $V_N(G)$ is $n(G) + N$, where $n(G)$ is called the order of G and N is the number of indeterminate nodes used in $V_N(G)$.

B. Degrees of truth, indeterminacy and falsehood

An edge represents a relationship in a neutrosophic semantic net being used to represent a sentence. The vertices represent words. Further, a vertex can be a noun, or an adjective, whereas an edge can be a verb or an adverb.

The membership of a vertex in a relationship can be either true or indeterminate or false. However, it can be ambiguous as well, when this membership is either both true and indeterminate, or both false and indeterminate. For example, in the sentence "He is probably a very good person", the membership of "He" is true to a certain degree as denoted by "very" and yet it is indeterminate to a certain degree as denoted by "probably". Therefore, the need for all three values *t, i* and *f* to be associated with each word in a sentence arises. To represent these three values, a vertex membership set of 3 values $\{t, i, f\}$ needs to be associated with each vertex. In this set the first element represents only degree of true

membership, the second element represents degree of indeterminate membership, and the third element represents degree of false membership, for each vertex *V*.

C. Degrees of positivity, neutrality and negativity

Further, neutrosophy is ambiguously used to simultaneously represent the triad concepts of truth, indeterminacy and falsehood, as well as, the triad concepts of positivity, neutrality and negativity. The term polarity is being introduced to distinguish the latter triad from the former.

Definition 14: Polarity

Polarity is defined as the term representing the polar concepts of positivity and negativity with a continuum of neutralities between the poles. It is a subset of neutrosophy.

Figure 1 shows the orientation of polarity with respect to crisp set theory, fuzzy theory and neutrosophy.

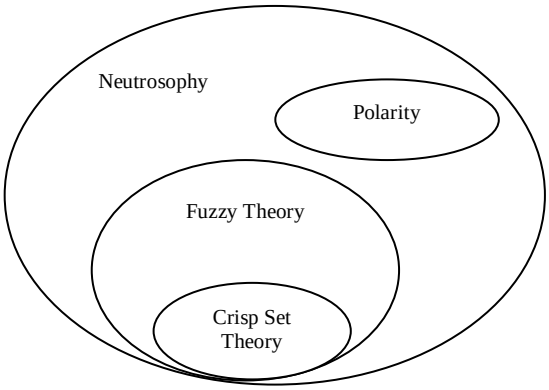


Fig. 1. Orientation of Polarity

1) Degrees of positivity, neutrality and negativity of a vertex

The three notions of truth, indeterminacy and falsehood can be substituted by the notions of positivity, neutrality and negativity, respectively. That is, the values *t, i* and *f* will be replaced by *p, u* and *n*, respectively, denoting the degree of positivity, the degree of neutrality and the degree of negativity. Thus, a polar vertex membership subset M_V is associated with each vertex *V*. Each M_V consists of 3 values (*p, u, n*). The set *M* is the polar vertex membership set associated with a graph *G* having a set of vertices *V*. Each vertex of *G* is then called as a polar neutrosophic vertex.

2) Degree of positivity, neutrality and negativity of an edge

Similar to the manner in which neutrosophic vertices were extended, neutrosophic edges need to be extended as well. That is, a set of three values (*p, u, n*) need to be associated with each edge.

Definition 15: Binary Neutrosophic Relation (Edges) [12]

A binary neutrosophic relation $R_N(X, Y)$ is a binary relation which assigns to each element of *X* two or more elements of *Y* or the indeterminate *I*. The notion of an adjacency matrix is associated with edges for denoting interconnections between them.

Definition 16: Adjacency matrix [11]

The edges *E* of an undirected graph *G* induce a symmetric binary relation *B* on *V* that is called the adjacency matrix of *G*. Specifically, for each edge $\{V_i, V_j\}$ the vertices V_i and V_j are said to be adjacent to one another.

Definition 17: Neutrosophic Matrix [12]

A neutrosophic matrix implies a matrix whose entries are from the set $N = [0, 1] \cup I$.

Definition 18: Fuzzy Neutrosophic Matrix [12]

A fuzzy neutrosophic matrix implies a matrix whose entries are from $N' = [0, 1] \cup \{nI \mid n \cup (0, 1)\}$.

Definition 19: Neutrosophic Adjacency Matrix [11]

Let *G* be a neutrosophic graph. The adjacency matrix of *G* with entries from the set (*I*, 0, 1) is called the neutrosophic adjacency matrix of the graph.

To represent these three values, a 3-dimensional neutrosophic adjacency matrix A_{ijk} is necessary, in which the first element in the third dimension A_{ij1} represents only *t* values, the second element A_{ij2} represents *i* values, and the third element A_{ij3} represents *f* values.

Definition 20: Fuzzy Neutrosophic Adjacency Matrix

Let *G* be a neutrosophic graph. The adjacency matrix of *G* with three entries (*t, i, f*) each from the set $N' = [0, 1] \cup \{nI \mid n \in (0, 1]\}$ for each edge of G is called a fuzzy neutrosophic adjacency matrix.

Definition 21: Polar Neutrosophic Adjacency Matrix

Let *G* be a neutrosophic graph. The extended adjacency matrix of G with three entries (*p, u, n*) each from the set $N = [0, 1] \cup I$ for each edge of *G* is called a polar neutrosophic adjacency matrix.

Definition 22: Polar Fuzzy Neutrosophic Adjacency Matrix

Let *G* be a neutrosophic graph. The extended adjacency matrix of *G* with three entries (*p, u, n*) each from the set $N' = [0, 1] \cup \{nI \mid n \in (0, 1]\}$ for each edge of *G* is called a polar fuzzy neutrosophic adjacency matrix.

IV. Polar Fuzzy Neutrosophic Semantic Net

Definition 23: Fuzzy Neutrosophic Semantic Net (FNSN)

A FNSN is defined as a strongly neutrosophic graph $G(V, E)$ in which a fuzzy neutrosophic membership set M^F is associated with *V* and a fuzzy neutrosophic adjacency matrix A^F is associated with *E*.

Definition 24: Polar Neutrosophic Semantic Net (PNSN)

A PNSN is defined as a strongly neutrosophic graph $G(V, E)$ in which a polar neutrosophic membership set M^P is associated with V and a polar neutrosophic adjacency matrix

A^P is associated with *E*.

Definition 25: Polar Fuzzy Neutrosophic Semantic Net (PFNSN)

A PFNSN is defined as a strongly neutrosophic graph *G (V, E)* in which a polar fuzzy neutrosophic membership set M^U is associated with *V* and a polar fuzzy neutrosophic adjacency matrix A^U is associated with *E*. A summary of the above terminology is given in Table 1.

A. A Polar Fuzzy Neutrosophic Semantic Net as an extended Semantic Net

The correspondence shown in Table 2, between the mathematical concepts underlying a semantic net and those underlying a PFNSN, clearly imply that a PFNSN is an extension of a semantic net.

The difference between a PFNSN and a traditional semantic net (TSN) is that a TSN cannot represent the concepts of neutrosophy and polarity. This is a serious drawback of a TSN since most of the sentences that human beings use in various situations are either neutrosophic or polar or polar neutrosophic.

B. Illustration of the representation of a sentence using a Fuzzy Neutrosophic Semantic Net

Consider the sentence S1: "The night is rather cold and somewhat hazy but it is not raining".

TABLE I. SUMMARY OF TERMINOLOGY

Semantic Net type	Membership Vertex set	Adjacency Matrix
Fuzzy Neutrosophic Semantic Net	Fuzzy Neutrosophic Membership Vertex Set	Fuzzy Neutrosophic Adjacency Matrix
Polar Neutrosophic Semantic Net	Polar Neutrosophic Membership Vertex Set	Polar Neutrosophic Adjacency Matrix
Polar Fuzzy Neutrosophic Semantic Net	Polar Fuzzy Neutrosophic Membership Vertex Set	Polar Fuzzy Neutrosophic Adjacency Matrix

TABLE II. CORRESPONDENCE BETWEEN THE MATHEMATICAL CONCEPTS UNDERLYING A SEMANTIC NET AND THE MATHEMATICAL CONCEPTS UNDERLYING A PFNSN

Concepts of Linguistics	Equivalent Mathematical Concepts of a Semantic Net	Equivalent Mathematical Concepts of a Polar Fuzzy Neutrosophic Semantic Net
Semantic Net	Graph	Neutrosophic Graph
Concept or object	Node or vertex	Polar Fuzzy Neutrosophic Vertex
Degree of participation in a relationship	Not Represented	Degree of positivity, neutrality and negativity of the membership of a vertex in a relationship-Polar Fuzzy Neutrosophic Membership Vertex Set
Relationship	Binary Relation (Edges)	Binary Neutrosophic Relation (Edges)
Degree of Polarity	Not Represented	Degree of positivity, neutrality and negativity of the relationship between two vertices
Semantics of a sentence	Adjacency Matrix or Connection Matrix	Polar Fuzzy Neutrosophic Adjacency Matrix

S1 is a moderately complex sentence which describes the weather at night. The weather is rather cold (truth *t* to a degree of 2.4), somewhat hazy (indeterminate *i* to a degree of 1.4) and not raining (false *f* that it is raining to a degree of 1.0). Thus this sentence exhibits neutrosophy (*t, i, f*) but the degrees of *t, i* and *f* are fuzzy. Hence the sentence is fuzzy neutrosophic and can be sufficiently represented by a FNSN. Table 3 shows the elements of S1 required for creating the FNSN for S1. Implementation of an FNSN was done in MATLAB and S1 was represented as shown in Figure 2.

TABLE III. THE ELEMENTS OF THE FNSN FOR THE SENTENCE S1.

Input/ Output	Mathematical Concepts of a Fuzzy Neutrosophic Semantic Net	Elements of the sentence S1
Inputs	Fuzzy Neutrosophic Vertices	night, cold, hazy, raining
	Degrees of truthhood(t), indeterminacy(i) and falsehood(f) of the membership of a vertex in a relationship- Fuzzy Neutrosophic Membership Vertex Set	$\begin{bmatrix} & t & i & f \\ night & 3.0 & 0 & 0 \\ cold & 3.0 & 0 & 0 \\ hazy & 3.0 & 0 & 0 \\ raining & 0 & 0 & 1.0 \end{bmatrix}$
	Binary Neutrosophic Relation (Edges)	rather(2.4), somewhat(1.4), not(1.0)
	Degrees of truthhood(t), indeterminacy(i) and falsehood(f) of the relationship between two vertices	Represented using a fuzzy neutrosophic adjacency matrix as in the next row of this table.
	Fuzzy Neutrosophic Adjacency Matrix A_{ij1} represents t, A_{ij2} represents i, A_{ij3} represents f	$A_{ij1}=\begin{bmatrix} 0 & 2.4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ $A_{ij2}=\begin{bmatrix} 0 & 0 & 1.4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ $A_{ij3}=\begin{bmatrix} 0 & 0 & 0 & 1.0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
Output	Neutrosophic Graph	See Figure 2

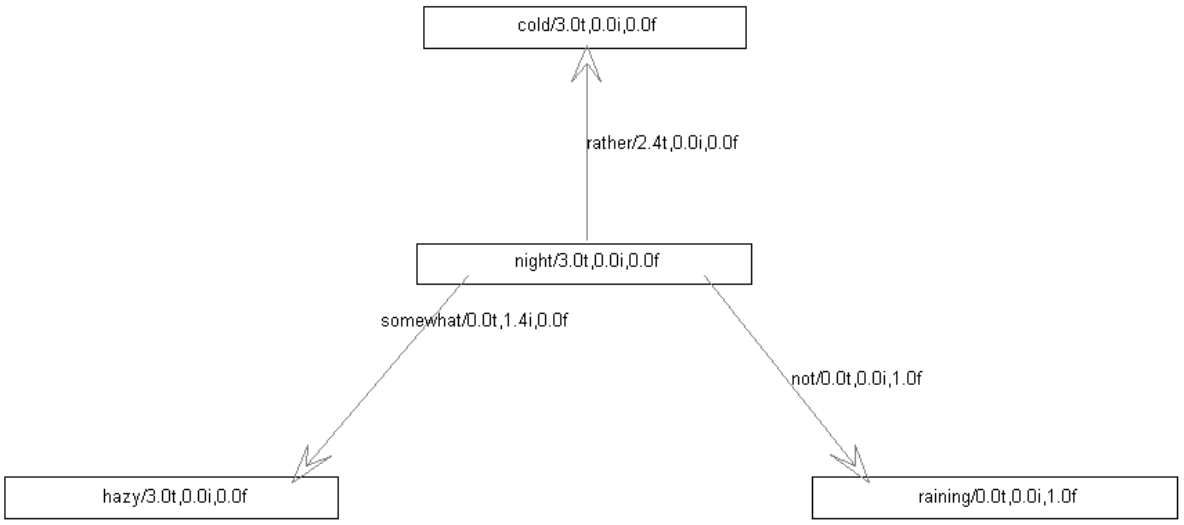


Fig. 2. Representation of the sentence S1 using MATLAB.

C. *Illustration of the representation of a sentence using a Polar Neutrosophic Semantic Net*

Consider the sentence S2: "An atom has protons, neutrons and electrons, where protons are positive, neutrons are neutral and electrons are negative".

S2 is a complex sentence which describes the components of an atom. These components, namely, protons, neutrons and electrons, are either completely part of an atom or completely not part of an atom, on the one hand, but are positively charged, neutral or negatively charged on the other. The charges represent the concept of polarity. But since the membership is complete in all cases, S2 does not exhibit fuzziness. Hence the sentence is polar neutrosophic.

Table 4 shows the elements of S2 required for creating the PNSN for S2. Implementation of a PNSN was done in MATLAB and S2 was represented as shown in Figure 3.

TABLE IV. THE ELEMENTS OF THE PNSN FOR THE SENTENCE S2.

Input/ Output	Mathematical Concepts of a Polar Neutrosophic Semantic Net	Elements of the sentence S2
Inputs	Polar Neutrosophic Vertices	protons, positive, neutrons, neutral, electrons, negative, atom
	Positivity(p), neutrality(u) and negativity(n) of a vertex taking part in a relationship-Polar Neutrosophic Membership Vertex Set	$\begin{bmatrix} p & u & n \\ 3.0 & 0 & 0 \\ 3.0 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 1.0 \\ 0 & 0 & 1.0 \\ 0 & 2.0 & 0 \end{bmatrix}$
	Binary Neutrosophic Relation (Edges)	are, has, are, has, are, has
	Positivity(p), neutrality(u) and negativity(n) of the relationship between two vertices	Represented using a polar neutrosophic adjacency matrix as in the next row of this table.
	Polar Neutrosophic Adjacency Matrix A_{ij1} represents p, A_{ij2} represents u, A_{ij3} represents n	$A_{ij1} = \begin{bmatrix} 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \end{bmatrix}$ $A_{ij2} = \begin{bmatrix} 0&2.0&0&0&0&0&2.0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&2.0&0&0&2.0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&2.0&2.0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \end{bmatrix}$

		$A_{ij3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
Output	Neutrosophic Graph	See Figure 3

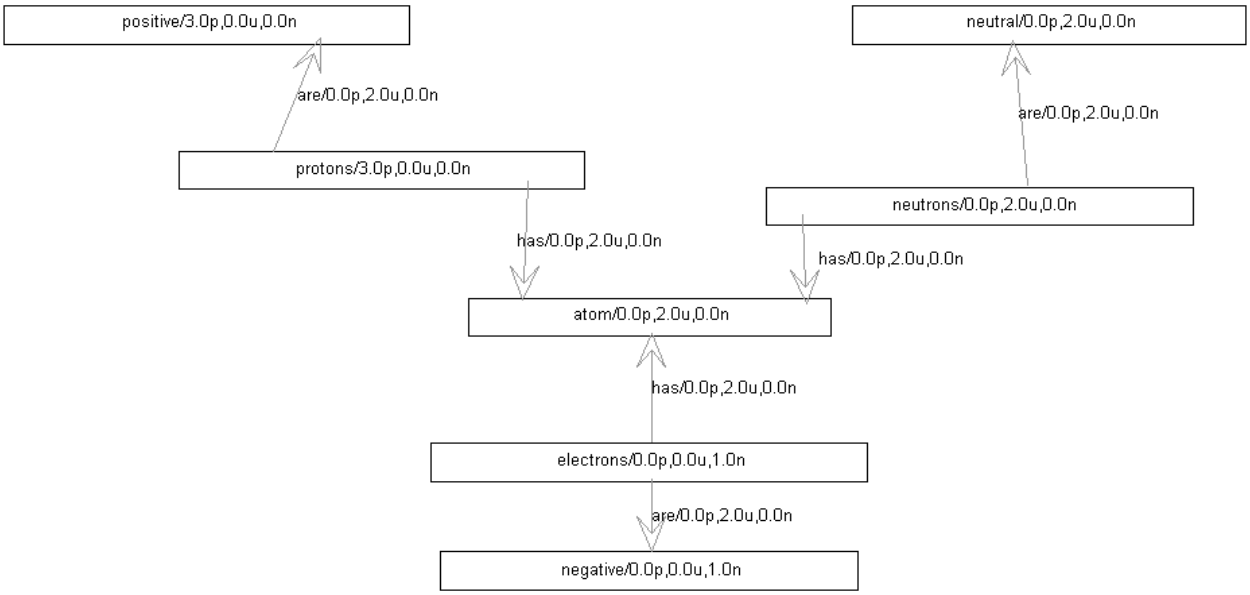


Fig. 3. Representation of the sentence S2 using MATLAB.

TABLE V. THE ELEMENTS OF THE PFNSN FOR THE SENTENCE S3.

Input/ Output	Mathematical Concepts of a Polar Fuzzy Neutrosophic Semantic Net	Elements of the sentence S3
Inputs	Polar Fuzzy Neutrosophic Vertices	Bob is quite healthy, rather plump but slightly anaemic
	Degrees of positivity(p), neutrality(u) and negativity(n) of the membership of a vertex in a relationship-Polar Fuzzy Neutrosophic Membership Vertex Set	$\begin{bmatrix} & p & u & n \\ Bob & 3.0 & 0 & 0 \\ healthy & 3.0 & 0 & 0 \\ plump & 3.0 & 0 & 0 \\ anaemic & 0 & 0 & 1.0 \end{bmatrix}$
	Binary Neutrosophic Relation (Edges)	quite (2.7), rather(1.4), slightly(0.3)
	Degrees of positivity(p), neutrality(u) and negativity(n) of the relationship between two vertices	Represented using a polar fuzzy neutrosophic adjacency matrix as in the next row of this table.
	Polar Fuzzy Neutrosophic Adjacency Matrix A_{ij1} represents p, A_{ij2} represents u, A_{ij3} represents n	$A_{ij1} = \begin{bmatrix} 0 & 2.7 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ $A_{ij2} = \begin{bmatrix} 0 & 0 & 1.4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ $A_{ij3} = \begin{bmatrix} 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
Output	Neutrosophic Graph	See Figure 4

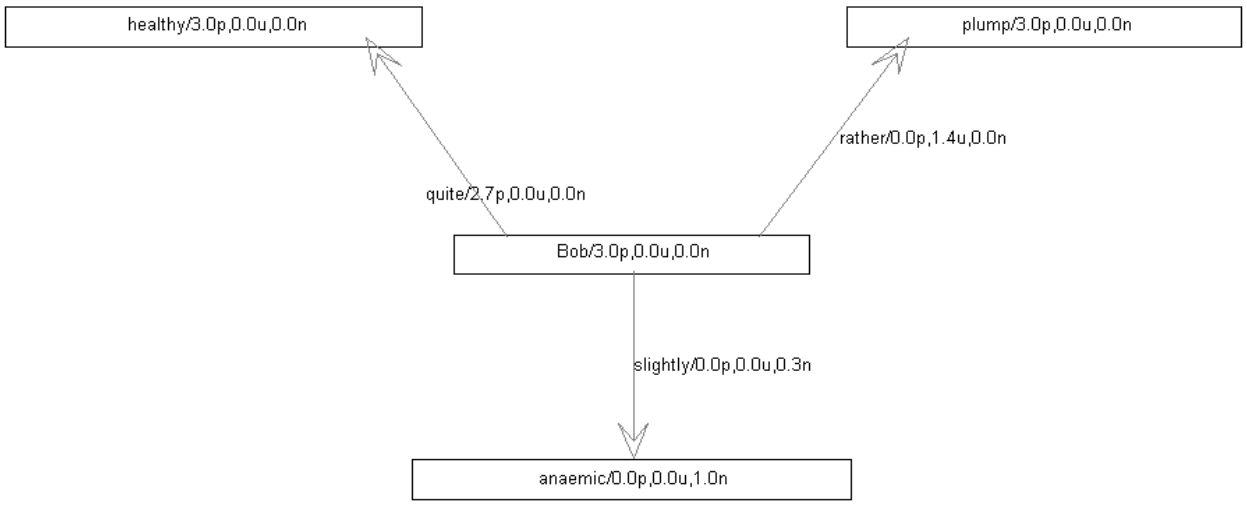


Fig. 4. Representation of the sentence S3 using MATLAB.

D. *Illustration of the representation of a sentence using a Polar Fuzzy Neutrosophic Semantic Net*

Consider the sentence S3: "Bob is quite healthy, rather plump but slightly anaemic".

S3 is a moderately complex sentence which describes the state of Bob's health. He is quite healthy (something positive but to a degree of 2.7), rather plump (something neutral to a degree of 1.4), but slightly anaemic (something negative, to a degree of 0.3, as indicated by the conjunction "but", which is creating an opposing notion). Since the degrees of polarity are fuzzy, therefore, the sentence is polar fuzzy neutrosophic.

Table 5 shows the elements of the sentence S3 required for creating the PFNSN for S3.

Implementation of a PFNSN was done in MATLAB and the above sentence was represented as shown in Figure 4.

V. Discussion

Each of the sentences S1, S2 and S3 have a basic structure which corresponds to a "picture" created by a human being in his mind when he creates a sentence for a given situation. Further, the structure, along with various degrees, defines the emotions with which a person utters the sentence as speech, after framing it. This lends the naturalness property to the voice of a person. The question of the applicability of fuzzy theory to speech processing has been explored in [13].

The fundamental idea underlying emotions is that they are positive, neutral or negative. A PFNSN can be directly applied for emotion representation in a machine. Further, since an intelligent response is characterized by answers that are based on an understanding of the positive and the negative aspects of the situation in which a sentence was spoken to a human being, a PFNSN can be used to generate responses which are nearer to being termed as "intelligent". A response can be considered intelligent if it takes into account positive as well as negative aspects of a situation. Aspects related to the applicability of fuzzy theory to intelligent response generation have been discussed in [14].

This can be done by decision making based on the polar selection of that vertex out of multiple vertices, which are connected to a given vertex.

VI. Conclusion

The incorporation of fuzziness, polarity and neutrosophy into a conventional semantic net, leading to a FNSN, a PNSN and a PFNSN is a major enhancement in terms of representational capability. This enhancement holds potential to incorporate emotion representation in an otherwise emotionless robot. This representation is to the extent of emotions being represented in the form of continua for each of the positive and negative poles and neutrality. This excludes the possibility of imbibing feelings, the physical outcome of emotion generation in a living being. Further, the speech of a robot can be imbibed with naturalness. As an extension of the speech, more humane expressions can be exhibited by a robot. Thus natural language processing using these extended semantic nets can form the core of developing human-like robots.

However, there are bound to be limitations on how well these semantic nets can represent emotions and the degree to which they can be scaled. On the other hand, this enhancement holds potential towards the development of more humane robots.

VII. Future Scope

The human mind receives inputs from the five senses. The inputs are in the form of images, video, audio, odour and touch. The inputs come together in the mind to form a picture of the current environment. This picture is the equivalent of an extended textual semantic net. These inputs are stimuli, to which the mind responds with emotions, which can be positive, neutral or negative, to qualify relationships between the inputs. Representation of these five inputs and incorporation of emotions into such a PFNSN is a promising future application area of the topic of this paper. This representation will form a part of the response subsystem of a robot.

REFERENCES

[1] E. Rich, and K. Knight, Artificial intelligence, Computer Science Series, McGraw-Hill, 1991.

[2] C. S. Peirce, Existential graphs, Collected Papers of Charles Sanders Peirce,1909.

[3] R. H. Richens, "General program for mechanical translation between any two languages via an algebraic interlingua," Report on research: Cambridge Language Research Unit, Mechanical Translation 3.2, 1956.

[4] R. R. Hightower, "Fuzzy semantic networks", Master's Thesis, Kansas State University, 1986.

[5] W. B. V. Kandasamy, and F. Smarandache, Fuzzy cognitive maps and neutrosophic cognitive maps, 2003.

[6] F. Smarandache, Neutrosophy / neutrosophic probability, set, and logic, American Research Press, 1998.

[7] F. Smarandache, A unifying field in logics: neutrosophic logic, neutrosophy, neutrosophic set, neutrosophic probability, American Research Press, Rehoboth, 2000

[8] N. Biggs, Encyclopedic dictionary of Mathematics, MIT Press, 1993.

[9] N. Biggs, Algebraic graph theory, Cambridge Mathematical Library, 2nd ed., Cambridge University Press, UK, 1993.

[10] F. Smarandache, "An introduction to neutrosophy, neutrosophic logic, neutrosophic Set, and neutrosophic probability and statistics," Proc. of the First Int. Conf. on Neutrosophy, Neutrosophic Logic, Neutrosophic Set, Neutrosophic Probability and Statistics, University of New Mexico, Gallup, USA, 1-3 Dec 2001.

[11] W. B. V. Kandasamy, and F. Smarandache, Basic neutrosophic algebraic structures and their applications to fuzzy and neutrosophic models, Vol. 4., Hexis, 2004.

[12] W. B. V. Kandasamy, and F. Smarandache, Fuzzy relational maps and neutrosophic relational maps, Hexis, Church Rock, 2004.

[13] S. Lakra, D. K. Sharma, T. V. Prasad, S. H. Atrey, "Application Of Fuzzy Mathematics To Speech-To-Text Conversion By Elimination of Paralinguistic Content", Proc. of Nat. Conf. on Soft Computing & Art. Intel. 2009, Lingaya's University, Faridabad, Jan 2009.

[14] T. V. Prasad, S. Lakra, G. Ramakrishna, "Applicability of Crisp and Fuzzy Logic in Intelligent Response Generation", Proc. of Nat. Conf. on Information, Computational Technologies and e-Governance (NCICTG) 2010, Alwar, Rajasthan, India, 19-20 Nov 2010, pp. 137-139.

Key Issues in Vowel Based Splitting of Telugu Bigrams

T. Kameswara Rao
Assoc. Professor and Head, CSE Dept
Brahma's Inst. of Engg. and Tech
Rajupalem, Nellore, AP, India

Dr. T. V.Prasad
Former Dean of Computing Sciences,
Visvodaya Technical Academy,
Kavali, AP, India.

***Abstract*—Splitting of compound Telugu words into its components or root words is one of the important, tedious and yet inaccurate tasks of Natural Language Processing (NLP). Except in few special cases, at least one vowel is necessarily involved in Telugu conjunctions. In the result, vowels are often repeated as they are or are converted into other vowels or consonants. This paper describes issues involved in vowel based splitting of a Telugu bigram into proper root words using Telugu grammar conjunction ('sandhi') rules for MT.**

***Keywords*—Telugu word splitting; vowel based splitting; compound word splitting; bigrams; trigrams; n-grams; NLP**

I. INTRODUCTION

Sanskrit is considered as the mother language for almost all Indian languages, since a majority of the Indian languages are based on grammar rules similar to that of Sanskrit grammar [6]. Sanskrit is grammatically very well structured and very rich in its inflections [7]. It is the oldest language on the earth to have a powerful structured grammar. Panini (300 BCE) the greatest grammarian developed Sanskrit grammar with more than 4000 rules [8], [10]. Unlike western languages, Sanskrit is the best example that unites the words to form a compound word (or simply compound). According to Bloomfield and Chomsky (1957), sentence is the largest grammatical unit [16].

There is a possibility and custom to write a complete sentence as a single compound in Sanskrit. For instance "*jalObhaumamantarikshamitidvidhAbhavati*" – for convenience, it can be tokenized as "*jalaH bhaumaM antarikshaM iti dvidhA bhavati*" means 'water is of two types, one is on the earth, and another is in space' ('*jalaH*' – water, '*bhaumaM*' – on the earth, '*antarikshaM*' – in space, '*iti*' – like this, '*dvidhA*' – two categories, '*bhavati*' – is).

Sanskrit scholars are to be very careful about tokenization. Lack of appropriate knowledge on the grammar or less attention to each and every letter gives immature tokenization that leads to yield distorted or quite opposite meaning in some special cases [8]. For example '*viSvAmitraH*' is the word to be tokenized; its meaning is friend of the universe. It can be tokenized as '*viSva' + 'amitraH*' according to '*savarNadIrga sandhi*', which is not to be applied here because it gives opposite meaning i.e., enemy of the universe. For this kind of special cases, Sanskrit gives exemptions strictly. So it should be '*viSva' + 'mitraH*', where regular conjunction rule is to be violated and special rule is applied. The person who is aware of this kind of special cases can only tokenize properly.

Likewise majority of Indian languages follow the features of Sanskrit; undergo conjunction which is inevitable that lead in generating compounds that are essentially bigrams, trigrams or n-grams. Bigram is a compound formed by two words and trigrams by three words, and so on. As Telugu is one of them, one can see the nature of uniting the words to form n-grams in Telugu also. Though Telugu is highly influenced by other languages, especially most of it is by Sanskrit [7], Telugu is not originated from Sanskrit [4]. Even though Telugu was originally intended to be totally free from Sanskrit, it has tremendous impact and deep penetration into Telugu. In 1816, Francis White Ellis raised this issue. Later Bishop Robert Cardwell proved that a family of twelve Dravidian languages Telugu, Tamil, Kannada/Canarese, Malayalam, Tulu, Kodagu/Coorg, Tuda, Kota, Gond, Khond/Ku, Rajmahal and Oraon are not originated from Sanskrit in his book titled "A Comparative Grammar of Dravidian Languages" in 1856 [5]. As a proof of that, pure Telugu literature work is available in the form of '*yayAti caritramu*' by '*ponnagaMTi telaganna*' written in 16th century [1][13]. Later Telugu mingled with Sanskrit heavily by '*samskrutAndhra kavulu*' (Sanskrit – Andhra - a synonym of Telugu - poets) when they translated epics in Sanskrit literature like Ramayana, Mahabharata and *bhAgavata*, etc. Learning or speaking Sanskrit was a great honor in those days and literature work in Sanskrit was highly honored. That can be one of the reasons to Sanskritize Telugu to enhance its value.

Additionally, there are numerous dialects in Indian languages - even many of them do not have script and are based on their culture, territory, and have tremendous impact of non-Indian languages like Urdu, Persian, Arabic, English, etc. For instance, most of the Telugu language is affected by Urdu in '*telaMgANa*' territory. '*tarfIdu, aafIsu, pennu, pEparu, kaburlu, bassu'*, etc. are words from those languages adapted in Telugu [4]. Such words, their conjunctions and their corrupted / colloquial forms are almost understandable by local humans but not easily by non-locals. For example '*nI jimmaDa*' is the word very frequently used by the natives of eastern Andhra. It means 'let your tongue fall' (literally 'jimma' is the colloquial form of '*jihva*' – Sanskrit word for tongue, '*aDa*' is the corrupted form of '*paDu*' – a Telugu word).

II. VOWELS

According to '*pANini*' Sanskrit grammar, vowels and their forms are given as in TABLE I [2] (pronunciations are given in Appendix).

TABLE I. CHARACTERISTICS OF VOWEL '*A* (अ)'

Vowel		Time required to pronounce	Types
Roman English	***dEva nAgari***		
a (short vowel)	अ (*hrasva*)	One unit (*Eka mAtra*)	*anudAtta, udAtta, svarita*
A (Long vowel)	आ (*dIrgha*)	Two units (*dvi mAtra*)	*anudAtta, udAtta, svarita*
*A*3 (Longer vowel)	आ3 (*pluta*)	Three units (*tri mAtra*)	*anudAtta, udAtta, svarita*

Note: '*pluta*' is applied in calling somebody who is at a distance. For example, '*hE rAmA*3'. Here '3' indicates the '*pluta*' of the vowel '*A*'. If '*pluta*' is not applied here, the person cannot be called.

Again each type is classified in to two different forms, namely '*anunAsika*' and '*ananunAsika*'. '*anunAsika*' is a nasal sound while '*ananunAsika*' is not. A total of six types for each vowel '*a, A, A*3' yields 18 different forms of vowel '*a*(अ)'. Likewise '*i*(इ), *R*(ऋ)' also have 18 different forms each. '*z*(ऌ), *E*(ए), *Y*(ऐ), *O*(ओ), and *W* (औ)' can be obtained in 12 forms for each as they are not derived long forms. A huge total of 132 vowels are there in Sanskrit. Mostly these are used in 'Vedas'. But only thirteen vowels '*a, A, i, I, u, U, R, Ru, z, E, Y, O* and *W*' are used in general usage. Two more vowels '*aM (anusvAra), aH (visarga)*' are used in Sanskrit. Two special vowels are there appears only in Sanskrit named '*jihvamUlIyam'* and '*upadhmAnIyam'*. If *'visarga'* is appeared as prior character of consonant '*k*', it is considered as '*artha-visarga'* and called '*jihvamUlIyam'*, e.g. '*aMtaHkaraNam'*. If '*visarga*' is appeared as prior character of consonant '*p*', it is called '*upadhmAnIyam*'. Ex. '*vAyuHpaMkam*'.

Telugu includes two more short vowels '*e*' and '*o*' and one more long vowel '*Z*' to the above listed Sanskrit vowels to comprise a total of eighteen vowels [2]. All proper Telugu words end with vowels only. That's why Telugu language is called '*ajanta*' (= '*ach*' + '*anta*', literally '*ach*' meaning vowel and '*anta*' meaning ending) language. Consonants are called '*hal*' in Telugu. They are 37 in number. Unlike Telugu, words of almost all Indian languages end in consonants and hence called '*halanta*' languages. All western languages are also categorized as '*halanta*' languages as their words commonly end in consonants except Italian that ends in vowels. This is the reason why Telugu is called 'Italian of the East' and one of the secrets behind sweetness of Telugu vocabulary.

There are eighteen vowels in Telugu language as shown in TABLE II. All the vowels are called '*ach*' or '*svara*' according to Telugu grammar, their Roman equivalents are as in TABLE II.

TABLE II. TELUGU VOWELS AND THEIR ROMAN EQUIVALENT

Telugu vowel	అ	ఆ	ఇ	ఈ	ఉ	ఊ	ఋ	ౠ	ఌ*
Roman English	*A*	*A*	*I*	*I*	*U*	*U*	*R*	*Ru*	*Z*
Telugu vowel	ౡ*	ఎ	ఏ	ఐ	ఒ	ఓ	ఔ	అం	అః
Roman English	*Z*	*E*	*E*	*Y*	*O*	*O*	*W*	*aM*	*aH*

Note: *Vowels **'ఌ, ౡ** (*z, Z*)**'** are not used now-a-days, they are not considered in this paper.

In Telugu, vowels are classified into two types as follows [14].

- '*hrasvAs' – 'a, i, u, R, z, e'*
- *'dIrghAs' – 'A, I, U, Ru, Z, E, Y, O, W*'

'*dIrghAs*' again classified into two types as follows.

- '*vakrAs' – 'e, E, o, O'*
- *'vakratamAs' – 'Y, W'*

III. PROCESS OF SPLITTING WORDS

Due to many practical issues involved in maintaining a database with all combinations of compounds, it is better to maintain only standard or root words. Compounds of the source language are split to obtain the original words using reverse engineering in accordance to the conjunction ('*sandhi*') rules. This will make the morphological analysis easier.

Proper stemming and correcting of corrupted forms for splitting of n-grams into individual tokens is necessary for better understanding the context. This plays an important role in translation also whereas understanding is also a kind of translation. Splitting of compounds into root words is an important phase in NLP for the applications like MT [9]. Building a computational model to analysis natural language is the goal of NLP [15]. For MT from Telugu to any other language including Indian languages, one of the issues of dealing with source language words is that each word need to be stored in the database together with different suffixes/prefixes (also known as inflections) thus tremendously increasing the storage space. This is in the case like Telugu that has about 800 different dialects within the state of Andhra Pradesh. But most of the conjunctions are common and are computable. The best way to translate them is to split back into root words as they formed and then translate individual root words. Compounds are formed with two or more root words. While the root words can be retrieved from database, the inflections thus obtained needs serious focus. Inability to sufficiently handle the inflections may result in false word formations and distorted meaning. But mere splitting the compound may not give complete meaning all the time. To understand the meaning of a compound, first identify the meaning of components and then the relationship between them [11]. For instance, a compound '*rAmunitOkapirAju*' is formed by two words '*rAmunitO* + *kapirAju*'.

First word is inflected, and second word is a root word. '*rAmunitO*' literally means with '*rAma*' and '*kapirAju*' means Hanuman. If the inflection is not observed in the first word, it may be split as '*rAmunitOka*' (literally means the tail of '*rAma*') + '*pirAju*' (an absurd word), which gives a distorted meaning.

The scope of this paper is limited to deal with bigrams only for obtaining better MT and aims to propose solutions to the issues of vowel based splitting. Issues related to handling of different types of dialects and their corrupted forms have not been considered. More specifically, handling of compounds formed according to the grammar rules, and their splitting based on vowels together with certain special cases in Telugu have been discussed.

IV. Conjunction Rules

Splitting is a process opposite to the conjunction. Conjunction is called '*sandhi*' and splitting is called '*sandhi vicchEda*' in Sanskrit. Telugu also use the word '*sandhi*' to represent conjunction. At least two words are required for conjunction. First word is called '*pUrva pada*' and the second word is called '*para pada / uttara pada*' [12]. While most part of the word remains unchanged, technically, the actual '*sandhi*' occurs between two letters, i.e., '*pUrva svaram*'(last letter of the '*pUrva pada*') and '*para svaram*' (first letter of the '*para pada*') [1]. Telugu language adapted many of the '*sandhi*' rules from Sanskrit as it uses much grammar of Sanskrit in addition to its own grammar rules. Sanskrit grammar rules were adapted into Telugu since majority words of Telugu language were taken from Sanskrit. Sanskrit grammar describes three ways to form a '*sandhi*'. They are

- **'*Agamamu*'** (literally means coming in Sanskrit): one new letter comes according to '*sandhi*' rules, and is included between the conjunction characters, without removing any of them. Ex. '*mA' +' amma'* = *'mAyamma'*. '*A'* and '*a*' are involved in '*sandhi*', the new letter '*y*' is included between '*A*' and '*a*'. '*tut*' is introduced in '*tuDAgama*', '*dud*' in '*dhuDAgama*', '*jam*' in '*jamuDAgama*' and so on, are the examples of '*Agama sandhis*' in Sanskrit and '*yaDAgama, TugAgama, rugAgama*' etc. are the examples of '*Agama sandhis*' in Telugu.
- **'*AdESamu*'** (literally means rule in Sanskrit): one new letter replaces the two '*sandhi*' letters. Ex. '*parama' + 'ISvaruDu'* = '*paramESvaruDu'*. '*a*' and '*I*' are involved in '*sandhi'* and both are replaced with '*E*'. '*yaNAdESa, anunAsika*' etc., are the examples of '*AdESa sandhis*' in Sanskrit and '*pumpvAdESa, gasaDadavAdESa*' etc., are the examples of '*AdESa sandhis*' in Telugu.
- **'*EkAdESamu*':** one character of the '*sandhi*' letters are omitted and second one continues to exist in the compound. Ex. '*rAmuDu' + 'ataDu' = 'rAmuDataDu'*. '*u*' and '*a*' are involved in '*sandhi*', but letter '*u*' is dropped and only '*a*' is continued. '*savarNadIrgha, guNa, vRddhi*' etc., are the examples of '*EkAdESa sandhis*' in Sanskrit, and '*akAra, ukAra, ikAra sandhis*' are the examples in Telugu.

In Sanskrit, there are five important classifications of '*sandhis*'. They are 1) '*ach sandhi*', 2) '*hal sandhi*' 3) '*visarga sandhi*' 4) '*prakruti bhAva sandhi*' and 5) '*svAdi sandhi*' [1]. But only first three '*sandhis*' are used very frequently. '*ach*' and '*visarga sandhis*' works with vowels and '*hal sandhis*' works with consonants. '*sandhi*' classifications are given in TABLE III.

TABLE III. List Of Sanskrit '*sandhis*'

***sandhi* Type**	**Names**
'ach'	*savarNa dIrgha, guN, vRddhi, yaNAdESa, vAntAdESa, yAntAdESa, pUrva rUpa, para rUpa, avaJnAdESA*
'hal'	*Scutva, shTutva, jaStva, anunAsika, pUrva savarNa, para savarNa, chatva*
'*visarga*'	This '*sandhi*' shows six types of differences, but names are not given to them.

Though these three kinds of '*sandhis*' are used by Telugu as it is, they are treated as Sanskrit '*sandhis*'. Telugu defines around thirty '*sandhis*' (TABLE IV) according to its grammar. These Telugu '*sandhis*' fall under '*ach sandhis*', '*hal sandhis*' or work with both vowels as well as consonants [1].

TABLE IV. List Of Telugu '*sandhis*'

S.No	***'sandhi' name***	***S.No***	***'sandhi' name***
1	*ukAra (utva)*	*16*	*penvAdi*
2	*yaDAgama*	*17*	*AmrEDita*
3	*AkAra*	*18*	*muvarNalOpa*
4	*IkAra*	*19*	*paDvAdi*
5	*apadAdisvara*	*20*	*aligAgama*
6	*dvirukta TakAra*	*21*	*anukaraNa*
7	*TugAgama*	*22*	*visandhi*
8	*RugAgama*	*23*	*paMpavarNAdESa*
9	*gasaDadavAdESa*	*24*	*trika*
10	*saraLAdESa(druta)*	*25*	*lu-la-na-la*
11	*puMpvAdESa*	*26*	*dugAgama*
12	*pugAgama*	*27*	*allOpa*
13	*prAtAdi*	*28*	*nakArAdESA*
14	*nugAgama*	*29*	*mivarNalOpa*
15	*itvAdESa*	*30*	*ukAra vikalpa sandhi*

Though there are many Sanskrit and Telugu '*sandhis*', only some of them for vowel based splitting have been considered which are resulting in a vowel in compound (TABLE V) irrespective of they are classified as '*ach sandhi', 'hal sandhi*' or '*visarga sandhi*'. Some special cases are also discussed in this paper even they are involving a consonant.

TABLE V. List Of '*sandhis*' Results A Vowel In Compound

S.No	**'*sandhi*' name**	**Result vowel**	***S/T***
1	*savarNadIrgha*	*A,I,U,Ru*	S
2	*guNa*	*E,O,ar*	S
3	*vRddhi*	*Y, W*	S
4	*visarga*	*O, H*	S
5	*akAra, ikAra, ukAra*	*a,A,i,I,u,U,e,E,Y,o,O,W*	T
6	*yaNAdESa*	*y* + vowel	T
7	*jastva sandhi*	*g/j/D/d/b* + vowel	S
8	*dviruktaTakAra*	*TT* + vowel	T

*S – Sanskrit, T – Telugu

1) 'savarNa dIrgha sandhi': This results a vowel 'A/I/U/Ru' accordingly when one of the following (TABLE 6) pattern occurs.

Note: Pattern is the combination of '*purvasvara*' and '*parasvara*'

TABLE VI. ALL PATTERNS OF '*SAVARNADIRGHA SANDHI*'

S.No	Pattern	Res	Example
1	*a + a*	*A*	*phAla + aksha = phAl**A**ksha*
2	*a + A*	*A*	*rAma + Alayamu = rAm**A**layamu*
3	*A + a*	*A*	*pUjA + arhuDu = pUj**A**rhuDu*
4	*A + A*	*A*	*prajA + Anati = praj**A**nati*
5	*i + i*	*I*	*kavi + iMdruDu = kav**I**MdruDu*
6	*i + I*	*I*	*naMdi + ISvara = naMd**I**Svara*
7	*I + i*	*I*	*vANI + iMdra = vAN**I**Mdra*
8	*I + I*	*I*	*vasumatI + ISa = vasumat**I**Sa*
9	*u + u*	*U*	*su + ukti = s**U**kti*
10	*u + U*	*U*	*mRdhu + Uruvu = mRdh**U**ruvu*
11	*U + u*	*U*	*vadhU + unnati = vadh**U**nnati*
12	*U + U*	*U*	*vadhU + Uruvu = vadh**U**ruvu*
13	*R + R*	*Ru*	*pitR + RNamu = pit**Ru**Namu*
14	*R + Ru*	*Ru*	Examples are not given since no words start or end with 'Ru' in Telugu.
15	*Ru + R*	*Ru*	
16	*Ru + Ru*	*Ru*	

2) 'guNa sandhi': This results in a vowel 'E/O/ar' accordingly when one of the following (TABLE VII) pattern occurs.

TABLE VII. ALL PATTERNS OF '*GUNA SANDHI*'

S.No	Pattern	Res	Example
1	*a + i*	*E*	*bhUtala + itara = bhUtal**E**tara*
2	*a + I*	*E*	*svarga + ISuDu = svarg**E**SuDu*
3	*A + i*	*E*	*mahA + ikshu = mah**E**kshu*
4	*A + I*	*E*	*mahA + ISuDu = mah**E**SuDu*
5	*a + u*	*O*	*dAma + udara = dAm**O**dara*
6	*a + U*	*O*	*Nava + Uha = nav**O**ha*
7	*A + u*	*O*	*mahA + uttama = mah**O**ttama*
8	*A + U*	*O*	*mahA + UrU= mah**O**rU*
9	*a + R*	*ar*	*brahma + Rshi = brahm**ar**shi*
10	*A + R*	*ar*	*mahA + Rshi = mah**ar**shi*

3) 'vRddhi sandhi': This results a vowel 'Y/W' accordingly when one of the following (TABLE VIII) pattern occurs.

TABLE VIII. ALL PATTERNS OF '*VRDDHI SANDHI*'

S.No	Pattern	Res	Example
1	*a + E*	*Y*	*Eka + Eka = Ek**Y**ka*
2	*a + Y*	*Y*	*Sarva + YSvarya = sarv**Y**Svarya*
3	*A + E*	*Y*	*kAMtA + Eka = kAMt**Y**ka*
4	*A + Y*	*Y*	*mahA + YSvarya = mah**Y**Svarya*
5	*a + O*	*W*	*Eka + Oshadhi = Ek**W**shadhi*
6	*a + W*	*W*	*rAma + Wnnatya = rAm**W**nnatya*
7	*A + O*	*W*	*mahA + Odhana = mah**W**dhana*
8	*A + W*	*W*	*kAMtA + Wnnati = kAMt**W**nnati*

4) 'visarga sandhi': This has five rules of which only two are considered since these two rules results in a vowel 'O/H' accordingly when one of the following (TABLE IX) pattern occurs.

Rule1: when '*pUrva svara*' is '*aH*' and '*para svara*' is '*a/u/g/gh/j /jh/D/Dh/d/dh/n/b/bh/m/y/r/l/v/h*', then '*pUrva svara*' is replaced with '*O*' in the compound.

TABLE IX. ALL PATTERNS OF '*VISARGA SANDHI*- 1'

S.No	Pattern	Res	Example
1	*aH + a*	*O*	*saH + ahaM = s**O**haM*
2	*aH + u*	*O*	*vijayaH + ullAsa = vijay**O**llAsa*
3	*aH + g*	*O*	*tiraH + gamana = tir**O**gamana*
4	*aH + gh*	*O*	*manaH + ghana = man**O**ghana*
5	*aH + j*	*O*	*saraH + ja = sar**O**ja*
6	*aH + jh*	*O*	*manaH + jhari = man**O**jhari*
7	*aH + D*	*O*	*naraH + DiMbha = nar**O**DiMbha*
8	*aH + Dh*	*O*	*SivaH + Dhamar = Siv**O**Dhamar*
9	*aH + d*	*O*	*yaH + dEvaH = y**O**dEvaH*
10	*aH + dh*	*O*	*tapaH + dhana = tap**O**dhana*
11	*aH + n*	*O*	*yaSaH + nagara = yaS**O**nagara*
12	*aH + b*	*O*	*manH + buddhi = man**O**buddhi*
13	*aH + bh*	*O*	*manaH + duHkh = man**O**duHkh*
14	*aH + m*	*O*	*SiraH + maNi = Sir**O**maNI*
15	*aH + y*	*O*	*manaH + yaMtra = man**O**yaMtra*
16	*aH + r*	*O*	*rajH + rAgamu = raj**O**rAgamu*
17	*aH + l*	*O*	*jalaH + lahari = jal**O**lahari*
18	*aH + v*	*O*	*tapaH + vanaM = tap**O**vanaM*
19	*aH + h*	*O*	*manaH + hara = man**O**hara*

Note: In this case, '*visarga*' should be preceded by '*a*' else, this rule is not applicable. Ex. '*dhanuH' + 'chalanamu' = 'dhanuScalanamu'.*

Rule2: *H + k / kh / p / ph* gives '*visarga*' as it is in the compound (TABLE X).

TABLE X. ALL PATTERNS OF '*VISARGA SANDHI*-2'

S.No	Pattern	Res	Example
1	*H + k*	*Hk*	*tapaH + kaMpa = tapa**Hk**ampa*
2	*H + kh*	*Hkh*	*hariH + khaDga= hari**Hkh**aDga*
3	*H + p*	*Hp*	*dhanuH + puMja= dhanu**Hp**uMja*
4	*H + ph*	*Hph*	*SaSiH + phalamu = SaSi**Hph**alamu*

Note: For this rule any vowel can precede the '*visarga*' and that vowel appears in the compound with preceding character of '*visarga*'.

5) 'ukAra sandhi': If 'pUrva svara' is 'u' and 'paras vara' is a vowel, then 'u' is replaced by the vowel in result (TABLE XI).

TABLE XI. ALL PATTERNS OF '*UKARA SANDHI*'

S.No	Pattern	Res	Example
1	*u + a*	*a*	*iTlu + anenu = iTl**a**nenu*
2	*u + A*	*A*	*kAlu + ADu = kAl**A**Du*
3	*u + i*	*i*	*vADu + ippuDu = vAD**i**ppuDu*
4	*u + I*	*I*	*kAlu + IDcu = kAl**I**Dcu*
5	*u + u*	*U*	*nEDu + unnADu = nED**u**nnADu*
6	*u + U*	*U*	*mEmu + Ugamu = mEm**U**gamu*
7	*u + e*	*E*	*Enugu + ekku = Enug**e**kku*
8	*u + E*	*E*	*vAgu + EtAmu = vAg**E**tAmu*
9	*u + Y*	*Y*	*siddhamu + Y = siddham**Y***
10	*u + o*	*o*	*rAmuDu + okaDu = rAmuD**o**kaDu*
11	*u + O*	*O*	*ippuDu + Orpu = ippuD**O**rpu*
12	*u + W*	*W*	*tinu + WshadhaM= tin**W**shadhaM*
13	*u + M*	*M*	*ipuDu + aMtamu = ipuDa**M**tamu*

Note: if '*ukAra sandhi*' rule is applied to split '*vAgISuDu*', it becomes '*vAgu' + 'ISuDu*', which is a wrong splitting. It should actually be split as '*vAk' + 'ISuDu*'. Such conflicts should be handled carefully and may require manual checks.

6) 'akAra sandhi': This 'sandhi' has four rules but only one of them is considered since remaining results in a consonant.

Rule: when '*pUrva svara*' is 'a' and '*parasvara*' is any vowel, then '*a*' is replaced by the vowel in result (TABLE XII).

TABLE XII. ALL PATTERNS OF '*AKARA SANDHI'*

S.No	Pattern	Res	Example
1	*a + a*	*a*	*rAma + anna = rAm**a**nna*
2	*a + A*	*A*	*ciMta + Aku = ciMt**A**ku*
3	*a + i*	*I*	*puTTina + illu = puTTin**i**llu*
4	*a + I*	*I*	*cinna + Iga = cinn**I**ga*
5	*a + u*	*u*	*cUDaka + uMDu = cUDaku**M**Du*
6	*a + U*	*U*	*Kotta + Uyala = kott**U**yala*
7	*a + e*	*e*	*sIta + ekkaDa = sIt**e**kkaDa*
8	*a + E*	*E*	*tella + Enugu = tell**E**nugu*
9	*a + Y*	*Y*	*nava + YSvarya = nav**Y**Svarya*
10	*a + o*	*o*	*cIma + okaTi = cIm**o**kaTi*
11	*a + O*	*O*	*konta + Opika = kont**O**pika*
12	*a + W*	*W*	*maha + WnnatyaM= mah**W**nnatyaM*

7) 'ikAra sandhi': if 'pUrva svara' is 'i' and 'parasvara' is a vowel, then 'i' is replaced by the vowel in result (TABLE XIII).

TABLE XIII. ALL PATTERNS OF '*IKARA SANDHI*'

S.No	Pattern	Res	Example
1	*i + a*	*a*	*Emi + aMTivi = Em**a**MTivi*
2	*i + A*	*A*	*nallani + Avu = nallan**A**vu*
3	*i + i*	*I*	*vacciri + ipuDu = vacci**i**puDu*
4	*i + I*	*I*	*ciTTi + ItakAya = ciTT**I**takAya*
5	*i + u*	*u*	*idi + unnadi = id**u**nnadi*
6	*i + U*	*U*	*cakkani + Uru = cakkan**U**ru*
7	*i + e*	*e*	*idi + evaridi = id**e**varidi*
8	*i + E*	*E*	*Takkari + Enugu = Takkar**E**nugu*
9	*i + Y*	*Y*	*idi + YrAvatamu = id**Y**rAvatamu*
10	*i + o*	*O*	*nETiki + okkaTi = nETik**o**kkaTi*
11	*i + O*	*O*	*idi + Orugallu = id**O**rugallu*
12	*i + W*	*W*	*ciTTi + Wshadhi = ciTT**W**shadhi*

Note: There are some special issues in this 'sandhi', like 'cEsi' + 'ipuDu' = 'cEsi**yi**puDu', 'vacci' + 'iccenu' = 'vacci**yi**ccenu'.

V. VOWEL BASED SPLITTING RULES

Technically, whatever the rules used for conjunction, they are used in reverse order to obtain those root words back. This approach can be considered as a reverse engineering process.

Algorithm:

1) A compound in Telugu, which is to be translated, is taken and is transliterated into Roman Telugu.

2) Each character is checked to determine whether it is a vowel.

3) If it is a vowel, then try all possible combinations to split the word according to the 'sandhi' rules listed in Tables 6 through 13.

4) If the compound is formed according to 'sandhi' rules of two words, then it is split into two words.

5) The process is recursively processed till all the words thus separated are found in the dictionary/database.

Example: '*SivArcana*' – formed by the root words '*Siva' + 'arcana*'. While using vowel based splitting, the vowels of '*SivArcana*' i.e., '*i, A, a*' are to be checked (TABLE XIV).

TABLE XIV. POSSIBLE PATTERNS OF THIS SPLITTING OF '*SIVARCANA*'

V	Pattern	Sandhi	Result	NA/A
i	*u + i*	*ukAra*	*Su + ivArcana*	NA
i	*a + i*	*akAra*	*Sa + ivArcana*	NA
i	*i + i*	*ikAra*	*Si + ivArcana*	NA
A	*a + a*	*savarNadIrgha*	*Siva + arcane*	**A**
A	*a + A*	*savarNadIrgha*	*Siva + Arcana*	NA
A	*A + a*	*savarNadIrgha*	*SivA + arcane*	NA
A	*A + A*	*savarNadIrgha*	*SivA+ Arcana*	NA
A	*u + A*	*ukAra*	*Sivu + Arcana*	NA
A	*a + A*	*akAra*	*Siva + Arcana*	NA
A	*i + A*	*ikAra*	*Sivi + Arcana*	NA
a	*u + a*	*ukAra*	*SivArcu + ana*	NA
a	*a + a*	*akAra*	*SivArca + ana*	NA
a	*i + a*	*ikAra*	*SivArci + ana*	NA

Note: If V (vowel) is the last character of the compound, then there will be a chance of split when the letter is a long vowel like '*A, I, U, E, Y, O'* e.g., '*vaccADA' = 'vaccADu' + 'A*' (means, 'did he come?').

This is occurs almost in interrogative cases. But there is no chance for short vowels to be the result of conjunction. There is no need to check the last character of the compound, if it is '*a, i, u, R, e or o*' assuming it is the result of '*sandhi*'. From all the patterns listed in TABLE XIX, '*a + a*' pattern of '*savarNadIrgha sandhi*' is applicable to split '*SivArcana*' into '*Siva + arcana*'. Amongst these 13 patterns, only one pattern is suitable to split the compound properly.

When one pattern splits the compound successfully, then there is no need to go for further splitting until unless the compound is formed by three or more. Unnecessary splitting may yield improper or unacceptable root words. As a rule of thumb, best results are obtained by splitting in such a way that first word extracted from the compound is as long as possible. Even if a proper word is obtained from the compound much before finishing, splitting process is not to be stopped until all vowels of the compound are checked. Ex. '*adhikAramaDugu*' is a bigram formed by two proper words '*adhikAramu*' (authority) and '*aDugu*' (to ask) by the rule of '*ukAra sandhi*'. But it can also be assumed as a trigram formed by three proper words '*adhi*' (to overcome), '*kAramu*' (chilli powder), '*aDugu*' (to ask). If splitting process is stopped at the earlier stage when it found a proper word (for instance, '*adhi*'), it yields useless or distorted meaning when translated.

Sometimes, some words are not to be treated as compounds and should be translated as a whole. For instance, '*adhikAri*' literally meaning "officer" is the word to be treated as single word and should not be split. If it is split, it becomes,

'*adhika + ari*' by the pattern '*a + a = A*' from '*savarNadIrgha sandhi*'. '*adhika*' (means 'more') and '*ari*' (means 'enemy'). Both are root words and '*sandhi*' seems to be proper but the meaning yields 'more enemy', an incorrect translation. The primary requirement in translation is that the meaning of the context should not be disturbed.

VI. SPECIAL CASES OF '*ACH SANDHI'*

All the '*sandhis*' and the cases discussed above are related to single independent vowel. There are special cases in which either next or previous letters of the vowel is also to be checked in splitting. This ensures that the compound is formed by a particular '*sandhi*'.

For some '*sandhi*' rules, both the previous and next letters of the vowel are to be checked (TABLE XVIII). Following are the examples.

1) 'guNa sandhi': In specific cases, this 'sandhi' results in two letters instead of one in compound (TABLE VII). Sometimes more than one letter also to be checked since, to reduce time complexity in splitting i.e. six patterns causes to result in vowel 'a' but only three patterns can result 'ar'. Ex. For 'brahmarshi' – is a compound formed by two root words 'brahma' and 'Rshi'. All patterns are given in (TABLE XV, XVI).

TABLE XV. POSSIBLE SPLITTING BY OBSERVING ONLY VOWEL '*A*'

Pattern	*sandhi*	Split forms	Result	NA/A
a + a	*akAra*	*brahma + arshi*	Fail	NA
u + a	*ukAra*	*brahmu + arshi*	Fail	NA
i + a	*ikAra*	*brahmi + arshi*	Fail	NA
aH + part2	*visarga*	*brahmaH + rshi*	Fail	NA
a + R	*guNa*	*brahma + Rrshi*	Fail	NA
A + R	*guNa*	*brahmA + Rrshi*	Fail	NA

TABLE XVI. POSSIBLE SPLITTING BY OBSERVING TWO LETTERS '*AR*'

Pattern	*sandhi*	Split forms	Result	NA/A
a + R	guNa	brahma + Rshi	Succeeded	A
A + R	guNa	brahmA + Rshi	Fail	NA

From the Tables 15 and 16 it is observed that when a conjunction results in two or more letters, the total numbers of letters are to be observed for splitting.

2) 'yaNAdESa sandhi': Though 'yaNAdESa sandhi' is an 'ach sandhi', it results in generating a consonant in the compound along with the vowel.

Rule: When '*pUrva svara*' is '*i/u/R'* and '*para svara*' is '*i/u/R*' then, '*y*' replaces '*i', 'v*' replaces '*u*' and '*r*' + vowel replaces '*R*' as '*AdESam*' in the result (TABLE XVII).

Note: '*ya, va, ra*' are called '*yaNNs*' in Sanskrit grammar. When '*sandhi*' is formed, '*yaNNs*' comes as '*AdESam*'. That's why this '*sandhi'* is named '*yaNAdESa sandhi*'[3].

While checking vowels of the compound in vowel based splitting, if the vowel if preceded with the letter '*y/v/r*' then consider both the letters '*y/v/r'* + vowel to apply '*yaNAdESa sandhi*' rules in reverse engineering to find root words effectively.

TABLE XVII. ALL PATTERNS OF '*YANADESA SANDHI*'

S.No	Pattern	Res	Example
1	*i + a*	*ya*	*ati + aMta = at**ya**Mta*
2	*i + A*	*yA*	*gWri + Arcana = gWr**yA**rcana*
3	*i + u*	*Yu*	*ati + unnati = at**yu**nnati*
4	*i + O*	*yO*	*dadhi + OdanaM = dadh**yO**danaM*
5	*u + a*	*va*	*madhu + annamu = madh**va**nnamu*
6	*u + A*	*vA*	*guru + AJna = gur**vA**Jna*
7	*R + A*	*rA*	*pirR + Arjitamu = pit**rA**rjitamu*

3. '*jastva sandhi*': This '*sandhi*' also results in specific consonants along with vowels in compound. These specific 'consonant + vowel' patterns are helpful (Except in some cases - refer Note of '*ukAra sandhi*'), in tracing exactly the root words by applying '*jastva sandhi*' rules.

Rule: when '*pUrva svara*' is 'k/c/T/t/p' and '*parasvara*' is a vowel/ *g/j/D/d/b/h/y/v/r*' then, '*g/j/D/d/b*' come as '*AdESam*' (TABLE XVIII).

TABLE XVIII. VOWEL PATTERNS OF '*JASTVA SANDHI*'

S.No	Pattern	Res	Example
1	*k + I*	*gI*	*vAk + ISuDu = vA**gI**SuDu*
2	*c + a*	*Ja*	*ac + aMtamu = a**ja**Mtamu*
3	*T + a*	*Da*	*shaT + aMgamu = sha**Da**Mgamu*
4	*t + A*	*dA*	*sat + AcAramu = sa**dA**cAramu*
5	*p + a*	*ba*	*kakup + aMtamu = kaku**ba**Mtamu*
6	*t + E*	*dE*	*tat + Ekamu = ta**dE**kamu*
7	*t + I*	*dI*	*jagat + ISuDu = jaga**dI**SuDu*

3) 'dvirukta TakAra sandhi': Occurrence of 'T' two times is called 'dvirukta TakAramu'.

Note: '*dvi*' means two, '*ukta*' means to tell, '*TakAramu*' means the letter '*Ta*'.

Rule: If the words '*kuru, ciru, kaDu, niDu, naDu*' connected with a vowel, then the letters '*ru', 'Du*' are replaced with '*TT*' (TABLE XIX).

TABLE XIX. ALL PATTERNS OF '*DVIRUKTA TAKARA SANDHI*'

S.No	Pattern	Res	Example
1	*ru + e*	*TTe*	*ciru + eluka = ci**TTe**luka*
2	*ru + u*	*TTu*	*kuru + usuru = ku**TTu**suru*
3	*Du + a*	*TTa*	*kaDu + aluka = ka**TTa**luka*
4	*Du + i*	*TTi*	*naDu + illu = na**TTi**llu*
5	*Du + U*	*TTU*	*niDu + Urpu = ni**TTU**rpu*
6	*Du + A*	*TTA*	*kaDu + Ayata = ka**TTa**yata*
7	*Du + e*	*TTe*	*kaDu + edura = ka**TTe**dura*

While checking vowels of the compound in vowel based splitting, if the previous two letters of the vowel are '*TT*', then '*dvirukta TakAra sandhi*' rules are followed in reverse engineering to find out easily the root words.

4) 'visarga sandhi': This 'sandhi' also results in specific consonants along with vowels in some special cases. These 'consonant + vowel' patterns are helpful in tracing root words by applying 'visarga sandhi' rules.

Rule 1: when '*pUrva svara*' is '*H*' and '*para svara*' is '*S/sh/s*' then, '*S/sh/s*' come as '*AdESam*' respectively (TABLE XX).

TABLE XX. VOWEL PATTERNS OF '*VISARGA SANDHI*-1'

S.No	Pattern	Res	Example
1	(v)*H + S*	(v)*SS*	*tapaH + Sakti = tap**aSS**akti*
2	(v)*H + sh*	(v)*shsh*	*catuH + shashTi = cat**ushsh**ashTi*
3	(v)*H + s*	(v)*ss*	*manaH + sAkshi = man**ass**Akshi*

Note: 'v' stands for 'vowel'

Rule 2: when '*pUrva svara*' is '*H*', for '*para svara*' '*c/ch', 'AdESa*' is '*S*', for '*para svara*' '*T,Th*', '*AdESa*' is '*sh*', for '*para svara*' '*t,th*', '*AdESa*' is 's' respectively (TABLE XXI).

TABLE XXI. VOWEL PATTERNS OF '*VISARGA SANDHI*-2'

S.No	Pattern	Res	Example
1	(v)*H + c/ch*	(v)*sc/ch*	*manaH + calana= man**asc**alana*
2	(v)*H + T/Th*	(v)*shT/Th*	*ushaH + TaMka = ush**ash**TaMka*
3	(v)*H + t/th*	(v)*stth*	*manaH + tApam = man**ast**Apam*

Note: 'v' stands for 'vowel'

While checking vowels of the compound in this splitting, if next two letters of the vowel are '*SS/shsh/ss/sc/shT/st*', then applying '*visarga sandhi*' rules give better results in reverse engineering to find out the root words easily.

VII. DISCUSSIONS

a) Inflections:

Inflections are called '*vibhaktis*' which play an important role in Telugu grammar. In Telugu, inflections occur at the rear part of a word which leads in altering the original form of the root word. If the word is inflected then it is not possible to carryout splitting straightaway. All inflections must be separated and splitting is applied to obtain root words. Conjunction is possible not only with root words but also with inflected words.

For example, '*bhUmyAkASamutO*' is the compound which is inflected ('*tO*') at rear end. It is separated first and split rule is applied to obtain '*bhUmi' + 'AkASamu*' (TABLE XVII).

If '*pUrvapada*' is inflected and participated in conjunction, then it is difficult to find out root word. For example, in the sentence '*rAmuNNeMduku cUSAvu*' (why did you see Rama?) the compound is '*rAmuNNeMduku*' and is to be split. It is known that this is formed by two words '*rAmuNNi' + 'eMduku*'. Since the first word is inflected ('*rAmuDu' + 'ni'*) and such words are not available in database as they are. In such cases, splitting should be applied using morphology rules.

b) Plural forms:

Plural forms are very common in any language. If they are involved in conjunction, splitting becomes difficult. For example, '*kukkalarupulu*' (barking of dogs) is a compound formed by '*kukkala' + 'arupulu*'. '*pUrva pada*' is a plural term and is inflected. Formation of some plural words is not proper. For example '*baLLu*' can be the plural form of either '*baDi*' (school), or '*baMDi*' (a cart). But '*baDulu*' and '*baMDlu*' are the right plural forms of '*baDi*' and '*baMDi*' respectively.

But in normal conversations, corrupted form '*baLLu*' is intermittently used for representing plurals for both. Likewise, '*paLLu*' can also act as plural form for '*paMDu*' (fruit) and '*pannu*' (teeth). '*paMDlu*' and '*paLLu*' are the plural forms of the above respectively. When these are involved in conjunction, splitting becomes much difficult. For example '*baLLunnavi*' is the compound formed by either '*baDi + lu + unnavi*' (schools are there) or '*baMDi' + 'lu' + 'unnavi*' (carts are there).

c) Colloquial forms:

Colloquial or corrupted form of language is inevitable. These corrupted forms become impossible to split until unless they are maintained in Database. For example, '*rAmoDoccADu*' (Rama came) is the compound formed by '*rAmuDu' + 'occADu*' where '*occADu*' is the corrupted form of '*vaccADu*'. For successful splitting, either '*occADu*' must also be available in database or a rule must be made to morph/consider it as '*vaccADu*'.

d) Problems caused by conjunctions:

Some conjunctions create difficulties in identifying the root verb. For example, '*mEDipaMDujUDu*' (see the fig fruit) is the compound of root words '*mEDipaMDu' +' cUDu*'. But according to conjunction rule ('*saraLAdESa sandhi'*) they must be '*mEDipaMDunu' + 'cUDu*' before conjunction. Here '*nu*' of the '*pUrva pada*' is removed and '*c*' of the '*parapada*' is converted to '*j*' and '*jUDu*' is not available in database.

'*gasaDadavAdESa sandhi*' also creates similar difficulties in splitting. For example, '*tallidaMDrulu*'(parents) is the compound formed by '*talli' + 'taMDrulu*'. According to conjunction rule, first letter of the '*parapada*' is converted from '*t*' to '*d*' and '*daMDrulu*' is not available in database. If '*da*' is morphed to '*ta*' for splitting, it leads to another difficulty. For example, '*Akalidappikalu*' (hunger and thirst) is the compound formed by '*Akali' + 'dappikalu*' by '*gasaDadavAdESa sandhi*'. If '*da*' of this compound is changed, then it becomes '*tappulu*' (mistakes) thus providing wrong translation.

VIII. CONCLUSION

Though there are many issues involved in splitting, splitting plays key role in MT. It paves a way to translate the source language as much as possible. Issues involved in the splitting can be solved by applying appropriate properly evolved morphological processes.

All possible patterns to observe in a compound for vowel based splitting given in TABLE XXII. Applying longest pattern as much as possible gives good results. Apply the rule of appropriate '*sandhi'* for splitting. When there are no multi-letter patterns available in compound, then it becomes mandatory to observe only single vowel for splitting. This may lead ambiguity in some cases. However, Vowel based splitting can separate at least one proper word from compound from left to right, if any. One can find more patterns for some special cases and can be included to split the compound very precisely.

TABLE XXII. TWO/THREE-LETTERS TO OBSERVE IN THIS SPLITTING

S.No	V	If Nx / Pr	Pattern	Split pattern	*Sandhi*
1	*a*	is – *r*	*Ar*	*a + R*	*guNa*
2	*a*	is – *r*	*ar*	*a + Ru*	*guNa*
3	*a*	is – *r*	*ar*	*A + R*	*guNa*
4	*a*	is – *r*	*ar*	*A + Ru*	*guNa*
5	*a*	is – *r*	*ar*	*aH*+ part2	*visarga*
6	*i*	is – *r*	*ir*	*iH* + part2	*visarga*
7	*I*	is – *r*	*Ir*	*IH* + part2	*visarga*
8	*u*	is – *r*	*ur*	*uH*+ part2	*visarga*
9	V	is – *y*	*y*+ V	*i* + vowel	*yaNAdESa*
10	*a*	is – *v*	*va*	*u + a*	*yaNAdESa*
11	*A*	is – *v*	*vA*	*u + A*	*yaNAdESa*
12	*A*	is – *v*	*vA*	*U + A*	*yaNAdESa*
13	*a*	is – *r*	*ra*	*R + a*	*yaNAdESa*
14	V	is – *g*	*g*+ V	*k* + V	*Jastva*
15	V	is – *j*	*j* + V	*c* + V	*Jastva*
16	V	is – *D*	*D* + V	*T* + V	*Jastva*
17	V	is – *d*	*d*+ V	*t* + V	*Jastva*
18	V	is – *b*	*b*+ V	*p* + V	*Jastva*
19	V	is – *TT*	*TT* + V	*Du/ru* + V	*dviruktTakAr*
20	V	is – *SS*	V + *SS*	V+*H* + *S*	*visarga*
21	V	is – *shsh*	V + *shsh*	V+*H*+*sh*	*Visarga*
22	V	is – *ss*	V + *ss*	V+*H*+*s*	*visarga*
23	V	is – *sc*	V + *sc*	V+*H* + *c*	*visarga*
24	V	is – *sch*	V + *sch*	V+*H* + *ch*	*visarga*
25	V	is – *shT*	V + *shT*	V+*H* + *T*	*visarga*
26	V	is – *shTh*	V + *shTh*	V+*H* + *Th*	*visarga*
27	V	is – *st*	V + *st*	V+*H*+*t*	*visarga*
28	V	is – *sth*	V + *sth*	V+*H*+*th*	*visarga*

* Here ‘pr’ for previous, ‘nx’ for next, and ‘V’ for vowel.

REFERENCES

[1] Malladi Krishna Prasad, “Telugu Vyaakaranamu”, Sri Venkateswara Book Depot, 2012.

[2] Dr. Samudrala Vemkata Ramga Ramanujacharya, “Samskruta Vaani” Rohini Publications, 1997.

[3] Kambhampati Ramagopala Krishnamurti, “Telugu Vyaakaranamu”, Sri Sailaja Publications, 1991.

[4] A.H. Arden, “A Progressive Grammar of the Telugu Language”, 2nd Edition, Society for promoting Christian Knowledge, Madras, 1905.

[5] Robert Caldwell, “A Comparative Grammar of The Dravidian or South-Indian Family of Languages”, 2nd Edition, 1875.

[6] Akshar Bharati, Amba Kulkarni and V Sheeba, “Building a Wide Coverage Sanskrit Morphological Analyzer: A Practical Approach”, The First Nat. Symp. on Modelling and Shallow Parsing of Indian Languages, IIT Bombay, 2006.

[7] Malhar Kulkarni, Chaitali Dangarikar, Iravati Kulkarni, Abhishek Nanda, and Pushpak Bhattacharya, “Introducing Sanskrit Wordnet”, Proc. of Global Wordnet Conf. 2010, Mumbai, India, 2010.

[8] Arthur. A. McDonell, “Sanskrit Grammar for Students”, 3rd Edition, Oxford University Press, 1926.

[9] Joshi Shripad S. “Sandhi Splitting of Marathi Compound Words”, Int. J. on Adv. Computer Theory and Engg., Vol. 2 Issue 2, 2012

[10] S. Varakhedi, V. Jaddipal and V.Sheeba, “An Effort To Develop A Tagged Lexical Resource For Sanskrit”, FISSCL Paris, Oct 2007.

[11] Anil Kumar, Vipul Mittal, Amba Kulkarni “Sanskrit Compound Processor”, Sanskrit Computational Linguistics, pp. 57-69, 2010

[12] Shathaka Sagaram, available at http://shathakasagaram.blogspot.in/ 2011/05/blog-post_03.html

[13] Jasti Suryanarayana, “Sanskrit for Telugu Students” , Sri Balaji Printers, Tirupati, 1993

[14] Divakrala Venkata Avadhani, “Telugu in Thirty Days”, Andhra Pradesh Sahithya Academy, 1976.

[15] Akshar Bharati, Vineet Chaitanya, Rajeev Sangal, “Natural Language Processing – A Paninian Perspective”, Prentice-Hall of India, 1994.

[16] T. Suryakanthi, Dr. S. V. A. V. Prasad and Dr. T. V. Prasad “Translation of Pronominal Anaphora from English to Telugu Language”, Int. J. of Adv. Computer Sc. App., Vol. 4 Issue 4, 2013.

APPENDIX

Pronunciations: The letters should be pronounced normally as in English, except when they are italicized. If so, follow the TABLE XXIII.

TABLE XXIII. ROMAN TELUGU PRONUNCIATION

RT	Usage as	RT	Usage as	RT	Usage as
a	a in – That	*R*	Ru in – Ruk	*o*	O in - Obey
A	a in - jack	*Ru*	roo in – roof	*O*	oa in - Roar
i	i in - His	*e*	e in - When	*W*	Ou in - out
I	Ea in- east	*U*	oo in – fool	*aM*	um in - sum
u	u in – Put	*E*	a in - Hate	*aH*	aH in – aH
U	oo in –fool	*Y*	I in - Ice		

*RT stands for Roman Telugu and the capitalized letters should be pronounced with greater emphasis on them.

Solving Semantic Problem of Phrases in NLP Using Universal Networking Language (UNL)

M. F Mridha
Department of CSE,
University of Asia Pacific,
Dhaka, Bangladesh

Aloke Kumar Saha
Department of CSE,
University of Asia Pacific,
Dhaka, Bangladesh

Jugal Krishna Das
Department of CSE,
Jahangirnagar University,
Savar, Dhaka, Bangladesh

***Abstract*—This paper largely deals with the Semantic problem and generation of semantic relations, which are difficult problems in the field of natural language processing. In this work we looked at it through the knowledge based perspective and language phenomena in terms of rules and dictionary features. The work is more focused on solving the problems of Bangla sentences, thereby improving the accuracy of analysis process. The problem needs, particularly, a knowledge intensive solution. We have used insights from linguistics, towards solving this problem. Also, the usefulness of automatic extraction of features for words in the dictionary becomes evident through the work.**

Keywords—Syntax and Semantic analysis; Bangla Root Word; Morphology and UNL; NLP

I. INTRODUCTION

Syntax as part of grammar is a description of how words grouped and connected to each other in a sentence. There is a good definition of syntax for programming languages: "… syntax usually entails the transformation of a linear sequence of tokens (a token is key to an individual word or punctuation mark in a natural language) into a hierarchical syntax tree". Later we will see that the same definition also can be used for NL. Main problems on this level are: part of speech tagging (POS tagging), chunking or detecting syntactic categories (verb, noun phrases) and sentence assembling (constructing syntax tree).

Semantics and its understanding as a study of meaning covers most complex tasks like: finding synonyms, word sense disambiguation, constructing question-answering systems, translating from one NL to another, populating base of knowledge. Basically one needs to complete morphological and syntactical analysis before trying to solve any semantic problem.

In this paper we present the Root Word analysis of Bangla Sentences for UNL system. The major components of our research works touches upon

1) Different types of Ambiguity that is caused by Bangla Root Word

2) UNL Expression of the Bangla Root word and

3) Bangla sentences analysis. In section 2 we describe the Bangla Sentence structure.

In sections 3 and 4, we present our main works that include all the above three components

II. PROBLEM DEFINITION

A. Structural Ambiguity

A word, phrase, sentence or other communication is called ambiguous if it can be interpreted in more than one way. If the ambiguity is because of a multiple meanings of a word, it is called lexical ambiguity. One type of ambiguity, called structural ambiguity, arises due to more than one possible structure for the sentence.

ছেলেটি ভাল গান গায়।

It has two English translation one is "The boy sings well " or " The boy sings good song" in the first sentence it means he has a good singing voice and in the second sentence it means that the meaning of his song is good.

B. Attachment ambiguity

This is a specific type of structural ambiguity in which a clause or a phrase has more than one possible association in the tree structure of the sentence of which it is a part. If the ambiguity is about the attachment of a clause then it is called clause attachment and if it is about attachment of prepositional phrase it is called prepositional phrase attachment. Depending on the site of attachment there are at least two possibilities, noun attachment or verb attachment. For example,

সে বই পড়ে। "means He reads a book"

সে হাটতে গিয়ে পড়ে গেল ।means "he fell during the walking"

here same word "পড়ে" has two different meaning.

TABLE I. BANGLA AMBIGUOUS SENTENCE OF ROOT WORD "চল্"

Root word	Meaning when used in sentence	Relation	Example	
[চল্]	Leave	Agt,obj,gol,pur	সে কথা শেষ না করেই চলে গেল।	He has left the place not to finish the word.
	Continuity	Pos,obj,man	তার ব্যবসা ভালই চলছে।	His business is being run well.
	Go	Rec,plt,obj	চল গ্রামে যাই।	Let's go to village.
	Obey	Obj,mod	পিতামাতার	Obey the

			কথা মেনে চলা উচিত।	word of the parents.
	Invalid	Qua,obj,plc	সব নিয়ম এখানে চলে না।	All kinds of rules will not be implemented here.
	Tradition	Obj,mod	সেসবের চল এখন আর নেই।	The tradition of the said things are not being continued.
	Style	Pos,mod,aoj	তার চাল-চলন ঠিক সুবিধার না।	His living style is not satisfactory.
	Expiry	man	টাকাটা আর চলবে না।	Taka has been expired.
	Variable	obj	চলকের মান পরিবতিত হতে পারে।	The value of the driver may be changed.
	Recent	Obj,mod,tim	চলতি মাসে দাম আরও বাড়তে পারে।	The value may rise in this current month.

TABLE II. BANGLA AMBIGUOUS SENTENCE OF ROOT WORD "কহ্"

Root word	Meaning when used in sentence	Relation	Example	
[কহ্]	Say	Agt,nam	কহেন কবি কালিদাস।	The poet Kalidas has said.
	Option	Mod,aoj,obj	কহিবার কোন উপায় থাকল না।	There is no option to say.
	Repeat	Aoj,agt	কহিয়া কহিয়া আমি ক্লান্ত।	I have been tired by saying.
	Invite	plc	কহ্ কানে কানে।	Say in the ear.
	Cannot say	obj	কহিতে না পারি।	It cannot be said.
	speakable	Man,qua, aoj, pur	কহতব্য কত কিছুইতো ছিল।	There was so many things to speak.
	speakable	Aoj,rsn	কথাটি কহনযোগ্য নয়।	The word is not for speak.
	cannot say	obj	কথাটি কহা হইলো না।	The word has not been told.
	Said	Obj,agt,pos, scn,met	কথাটি কহিয়া মনে শান্তি পেলাম।	I have got the peace in my mind by saying
				the word.
	Flower	Aoj,pos,mod	কহলার আমাদের জাতীয় ফুল।	Lily is our national flower.

C. Why Semantic analysis of Root word need?

Verb is the main part of any sentence for any native language. Any Sentence can complete without subject or object. But without verb no sentence is complete. So verb analysis is need to converting from Bangla to UNL. And verb is the combination of root word and suffixes. And root word is titled as entry node when converting any native language to UNL. And not only verb but also other word is derived from a root word that may have the different transformations. This happens because different morphemes are added with it as suffixes. Therefore, the meaning of the word varies for its different transformations. We developed the following rules to this problem.

III. AMBIGUITY OF BANGLA WORD IN SENTENCES

It is necessary to make universal words in the context of bangle sentence and their usage. Converting to the English sentence and then make the universal word for Bangla language will not be semantically correct. In that case enconversion and deconversion will not be correct also.

If we check for some sentences like below:

ÒmK‡jB GK ¶z‡i gv_v gywo‡q‡Q †`LwQ, cix¶vq mevB †dj K‡i‡Q|Ó

Here, Ò¶z‡i gv_v gywo‡q‡QÓ – this clause means to have similar in nature. And in it is a type of Bagdhara. We have to make UW directly from Bangla sentence and it bear the correct meaning than. Both the enconversion and deconversion process will satisfy it.

On the other hand there should have some dictionary entries for special Bangla words which are called ÒGK K_vq cÖKvkÓ| This means to express a group of words in a single word shortly.

w`‡b †h GKevi Avnvi K‡i- GKvnvix|

‡Q‡jwU w`‡b GKevi Avnvi K‡i|

‡Q‡jwU GKvnvix|

Both the sentences are same in meaning. So, the Enconversion as well as Deconversion system should know this.

G Mv‡Qi wkKo Mfx‡i hvq|

here, root word [hv] 'go'and the immediate previous word [Mfx‡i]. It needs to make a relation / dependency checking among these two words as meaning of root depends on its previous word. So, there should have a technique of matching which retrieves appropriate word from the dictionary.

G Pv‡j †ekxw`b hv‡e bv|

here, root word [hv] and the immediate previous word [‡ekxw`b].

‡U^{a}b AvR hv‡e bv|

here, root word [hv] and the immediate previous word [AvR].

‡m ¯‹z‡j hvq|

here, root word [hv] and the immediate previous word [¯‹z‡j].

bZzb evwo‡Z K‡e hv‡eb?

here, root word [hv] and the immediate previous word [K‡e].

It needs to know semantically the use of root word in each sentence same root carry different meaning. For this it needs to analyze the words in the sentences before and after the main root/verb. The parser need to know which particular dictionary entry has to retrieve to make the universal word.

If the meaning of the root word is: [hv]- 'go'; - then the dictionary entry will be [hv]{}"go(icl>do)"(ROOT, BANJANT)

when it is of [hv]- spread than.

Rules for solving ambiguity of Bangla Root word[13][14][17].

Rule sets 1:

a) Some Suffixes (বিভক্তি) are used immediately after the root (ধাতু) for sadhu (সাধু) & cholito (চলিত) both languages.

b) Some Suffixes (বিভক্তি) are changed according to Person (পুরুষ) for sadhu (সাধু) & cholito (চলিত) both languages.

c) For sadhu (সাধু) the suffixes are [ϕ(শূন্য বিভক্তি),iya(ইয়া), ite(ইতে)] .

d) For cholito (চলিত) the suffixes are [ϕ(শূন্য বিভক্তি)and e (এ)] .

e) For Person (পুরুষ) the suffixes are [lam(লাম) , le(লে), l(ল), ch(ছ), che(ছে), chi(ছি), bo(ব), be(বে)] .

Rule sets 2: sadhu (সাধু) and **cholito (**চলিত) **language for different Tense (কাল)**

a) If the suffixes (বিভক্তি) for sadhu (সাধু) language is [ϕ(শূন্য বিভক্তি)] then the corresponding suffix (বিভক্তি) for cholito (চলিত) is [ϕ(শূন্য বিভক্তি)].

b) If the suffixes (বিভক্তি) for sadhu (সাধু) language is ite (ইতে) then the corresponding suffix (বিভক্তি) for cholito (চলিত) is [ϕ(শূন্য বিভক্তি)].

c) If the suffix (বিভক্তি) for sadhu (সাধু) language is iya (ইয়া) then the corresponding suffix for cholito (চলিত) is e [(এ)].

Rule set 3: Person (পুরুষ) (1st , 2nd and 3rd) (singular and *plural*)

a) If the suffixes (বিভক্তি) for 1st person is i(ই) then the corresponding suffix (বিভক্তি)for 2nd person and 3rd person is [ϕ(শূন্য বিভক্তি)] or O(ও) and e [(এ)] or y(য়) respectively.

b) If the suffixes (বিভক্তি) for 1st person is chi(ছি) then the corresponding suffix(বিভক্তি)for 2nd person and 3rd person is ch(ছ) and che(ছে) respectively.

c) If the suffixes (বিভক্তি) for 1st person is lam(লাম) or chilam (ছিলাম) then the corresponding suffix (বিভক্তি) for 2nd person and 3rd person is l(ল) or chil(ছিল) and le(লে) or chile(ছিলে) respectively.

d) If the suffixes (বিভক্তি) for 1st person is bo(ব) then the corresponding suffix (বিভক্তি) for 2nd person and 3rd person is be(বে) and be(বে) respectively.

e) If the suffixes (বিভক্তি) for 1st person is ai(আই) then the corresponding suffix (বিভক্তি)for 2nd person and 3rd person is ao(আও) and ay(আয়) respectively.

B. Dictionary Entry[12][15][7]

Shadhu Suffix

["^"]{} " " (PROT,KBIVOKTI,SHADHU,INDIFINIT)<B,0,0>
[ই]{} " " (PROT,KBIVOKTI,SHADHU,INDIFINIT)<B,0,0>
[ইতে]{}"^"
(PROT,KBIVOKTI,SHADHU,CONTINUOUS)<B,0,0>
[ইয়া]{}" " (PROT,KBIVOKTI,SHADHU,PERFECT)<B,0,0>

Cholito Suffix

["^"]{}" " (PROT,KBIVOKTI,CHOLITO,INDIFINIT)<B,0,0>
[" "]{}" "
(PROT,KBIVOKTI,CHOLITO,CONTINUOUS)<B,0,0>
[এ]{}" " (PROT,KBIVOKTI,CHOLITO,PERFECT)<B,0,0>

Person Suffix

[ই]{}" " (PROT,KBIVOKTI ,1P)<B,0,0>
[0]{}" " (PROT,KBIVOKTI, 2P)<B,0,0>
[ও]{}" " (PROT,KBIVOKTI, 2P)<B,0,0>
[এ]{}" " (PROT,KBIVOKTI, 3P)<B,0,0>
[য়]{}" " (PROT,KBIVOKTI, 3P)<B,0,0>
[ছি]{}" "(PROT,KBIVOKTI,1P)<B,0,0>
[ছ]{}" "(PROT,KBIVOKTI,2P)<B,0,0>
[ছে]{}" "(PROT,KBIVOKTI,3P)<B,0,0>
[লাম]{}" "(PROT,KBIVOKTI,1P)<B,0,0>
[লা]{}" "(PROT,KBIVOKTI,2P)<B,0,0>

[লে]{}“ ”(PROT,KBIVOKTI,3P)<B,0,0>
[ব]{}“ ”(PROT,KBIVOKTI,1P)<B,0,0>
[বে]{}“ ”(PROT,KBIVOKTI,2P)<B,0,0>
[বে]{}“ ”(PROT,KBIVOKTI,3P)<B,0,0>

IV. Our Proposed Method

There have two reasons to get better performance than other methods. The first reason is the effect of the approach in case of Bangla language. For example, the existing way to find the UNL expression uses three dimensions to find the enconverted or deconverted output which are i) Dictionary Entry Look-up, ii) Rules of morphological analysis and iii) Semantic Analysis. Since component nodes are created by using these steps, they may be less accurate sue which may not expresses semantically correct output as there are different language constraints. As a result, the converted expression may be grammatically correct one but not be meaningfully correct.

The second reason is the determination of the number of component nodes path for constructing desired output. Although the nature of input is meaningfully different as seen in Table-1 and Table 2, existing approach uses the grammatical attributes to select component nodes for all problems. However, Attribute Analysis Approach uses different options to get actual path in the component networks for different input sentence based on the nature of input.

Flow Chart of our proposed program Architecture is shown below:

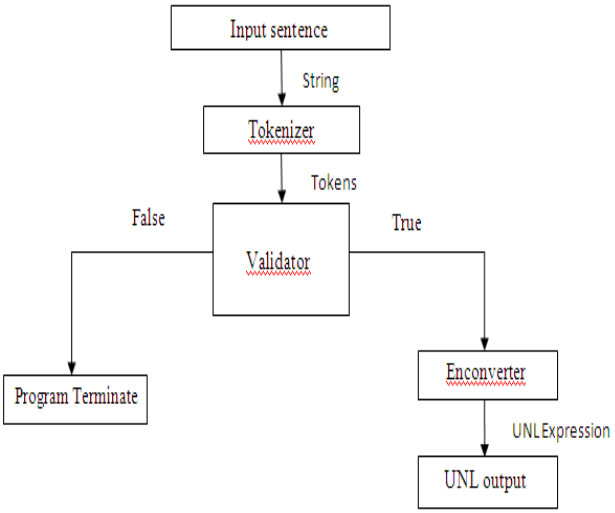


Fig. 1. Proposed program Architecture

Here,
Input sentence: The Bengali sentence which will be converted to UNL expression.
This sentence is given as string.

Example: “আমি ভাত খাই”

Tokenizer: In here the input sentence “String” is devide into tokens.

Example: “আমি” “ভাত” “খাই”
Token Token Token

Validator: It check that is the is the tokens are arranged in right order or check is there any grammatical mistakes in the given sentence or “String” or “Tokens”.

Example: “আমি তুমি খাই” - Invalid
“আমি ভাত খাই” – Valid

En-Converter: It convert the given sentence or “String” or “Tokens” in UNL expression.

Example: “আমি ভাত খাই”

UNL Expression:
{unl}

agt(eat(icl>consume>do,agt>living_thing,obj>concrete_thing,ins>thing).@entry.@present,i(icl>person))
obj(eat(icl>consume>do,agt>living_thing,obj>concrete_thing,ins>thing).@entry.@present,rice(icl>grain>thing))
{/unl}

V. Conclusion

Semantic Analysis Approach improves correct method of enconversion of UNL expression of Bangla language. A new technique has been proposed in this thesis work. The new technique used a constructive approach to determine the universal words of Bangla language. The novelty of this method is that, it used straightforward and simple technique to determine the ambiguity of Bangla word as well as the diversified usage of words in sentences for a given Bangla sentence. Semantic Analysis Approach first tried to solve the given problem by some example sentences, than it finds out required approaches to get semantically valid equivalence to get actual meaning of the sentence.

Semantic Analysis Approach explores a new era in universal word construction, i.e., determining number of paths by analyzing the dictionary entries; which leads to creates good options to find appropriate meaning of the input sentence for proper enconversion and deconversion process.

References

[1] H. Uchida, M. Zhu. The Universal Networking Language (UNL) Specification Version 3.0 Edition 3 ,Technical Report, UNU, (2005/6-UNDL Foundation, International Environment House, Tokyo, 2004)

[2] H. Uchida, M. Zhu, “The Universal Networking Language (UNL) Specification Version 3.0”, Technical Report, United Nations University, Tokyo, 1998

[3] S. Abdul-Rahim, A.A. Libdeh, F. Sawalha, M. K. Odeh, “Universal Networking Language(UNL) a Means to Bridge the Digital Divide”, Computer Technology Training and Indistrial Studies Center, Royal Scientific Sciety, March 2002.

[4] M. M. Asaduzzaman, M. M. Ali, “Morphological Analysis of Bengali Words for Automatic Machine Translation”, International Conference on Computer, and Information Technology (ICCIT), Dhaka, pp.271-276, 2003.

[5] Bengali Academy Bengali-English Dictionary, Dhaka (2004).

[6] Enconverter Specifications, version 3.3, UNL Center/ UNDL Foundation, Tokyo, Japan 2002.

[7] Enconverter Specification Version 3.3, (UNU Centre, Tokyo 150-8304, Japan 2002)

[8] DeConverter Specification, Version 2.7, (UNL Center, UNDL Foundation, Tokyo 150-8304, Japan 2002)

[9] D.M. Shahidullah. Bengali Baykaron, (Ahmed Mahmudul Haque of Mowla Brothers prokashani, Dhaka 2003)

[10] Zakir Hossain, Shahid Al Noor, Muhammad Firoz Mridha Some Proposed Standard Models for Bengali Dictionary Entries of Bengali Morphemes for Universal Networking Language. IJCSNS International Journal of Computer Science and Network Security, V OL.12 No.11, November 2012 .

[11] Bouguslavsky, I., Frid, N. and Iomdin, L. Creating a Universal Networking Module within an Advanced NLP system. Proceedings of the 18th International Conference on Computational Linguistics, pp. 83-89. (2000).

[12] Aloke Kumar Saha, Muhammad F. Mridha, Manoj Banik, and Jugal Krishna Das. Specification of UNL Deconverter for Bengali Language. International Journal of Scientific & Engineering Research, Volume 3, Issue 9, September-2012 ISSN 2229-5518.

[13] Muhammad Firoz Mridha, Md. Zakir Hossain, Shahid Al Noor, "Development of Morphological Rules for Bangla Words for Universal Networking Language" IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.10, October 2010.

[14] Muhammad Firoz Mridha, Kamruddin Md. Nur, Manoj Banik and Mohammad Nurul Huda, "Structure of Dictionary Entries of Bangla Morphemes for Morphological Rule Generation for Universal Networking Language". International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM) 2011.

[15] Muhammad Firoz Mridha, Kamruddin Md. Nur, Manoj Banik and Mohammad Nurul Huda, "Generation of Attributes for Bangla Words for Universal Networking Language(UNL)". International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM) 2011.

[16] Md. Sadequr Rahman, Sangita Rani Poddar, Muhammad Firoz Mridha, Mohammad Nurul Huda, "Open Morphological Machine Translation: Bangla to English". NWESP, page, 460-465 , ISBN: 978-1-4244-7817-0, 2010.

[17] Muhammad Firoz Mridha, Md. Nawab Yousuf Ali, Manoj Banik3, Mohammad Nurul Huda,Chowdhury Mofizur Rahman, Jugal Krishna Das, "Conversion of Bangla Sentences to Universal Networking Languages, " SKIMA'10, Paro, Bhutan, August 2010.

Analyzing Opinions and Argumentation in News Editorials and Op-Eds

Bal Krishna Bal
Information and Language Processing Research Lab
Department of Computer Science and Engineering
Kathmandu University, P.O. Box – 6250
Dhulikhel, Kavre, Nepal

***Abstract*—Analyzing opinions and arguments in news editorials and op-eds is an interesting and a challenging task. The challenges lie in multiple levels – the text has to be analyzed in the discourse level (paragraphs and above) and also in the lower levels (sentence, phrase and word levels). The abundance of implicit opinions involving sarcasm, irony and biases adds further complexity to the task. The available methods and techniques on sentiment analysis and opinion mining are still much focused in the lower levels, i.e., up to the sentence level. However, the given task requires the application of the concepts from a number of closely related sub-disciplines – Sentiment Analysis, Argumentation Theory, Discourse Analysis, Computational Linguistics, Logic and Reasoning etc. The primary argument of this paper is that partial solutions to the problem can be achieved by developing linguistic resources and using them for automatically annotating the texts for opinions and arguments. This paper discusses the ongoing efforts in the development of linguistic resources for annotating opinionated texts, which are useful in the analysis of opinions and arguments in news editorials and op-eds.**

Keywords—editorials; opinions; arguments; persuasion; sentiment analysis; annotation; NLP

I. Introduction

News editorials and op-eds, which fall under particular kinds of persuasive texts, are rich sources for discourse analysis on particular events. However, in the context of the growing number of news editorials both in the print and online media, such an analysis becomes difficult owing to at least two reasons – the first one being the enormous amount of content to handle and the other one being the challenge to decide on the relative biases and objectivity of the editorial texts. Since editorials are necessarily views and opinions of the news agencies or the columnist involved, it is often the case that all possible measures of persuasion are employed lest the text sounded convincing or persuading. It is quite a common phenomenon in such texts to come across opinions seemingly to be facts (opinions in disguise of facts), rhetoric, exaggerations, sarcasm and irony.

Given a computational perspective to address the above task, there is clearly a need to analyze the texts in different levels – the discourse level (paragraph level or above), the sentence level, phrase level and the word level.This encompasses the application of the concepts from a number of closely related disciplines like Sentiment Analysis, Argumentation Theory, Discourse Analysis, Computational Linguistics, Logic and Reasoning etc.[1]. Apparently, this is a difficult task for humans, let alone the machine. The primary argument of this paper is that partial solutions to the problem can be achieved by developing linguistic resources and using them for automatically annotating data for opinions and arguments. Such annotated data would be very useful in the analysis of opinions and arguments. This paper discusses the ongoing efforts in the development of linguistic resources for analyzing opinions and arguments in news editorials and op-eds.

The paper is organized in altogether seven sections. Section II introduces the underlying argument structure in persuasive texts. Section III talks about the current efforts made by the given research work in building a corpus of editorials and op-eds. Section IV explains the semantic tagset developed for annotating the corpus. Section V gives an overview of the different linguistic resources required for the annotation work. Section VI presents and discusses the results of the annotation work and performance of the automatic annotation tool. Finally, Section VII discusses the conclusion and future extensions to the given research work.

II. The Underlying Argument Structure In Persuasive Texts

Persuasive writings in general and particularly editorials of argumentation and persuasion exhibit the following argumentation structure[1]:

- Opening or thesis statement
- Support statements (facts/opinions)
- Conclusion

The opening or thesis statement introduces the issue or the problem in consideration while the support statements try to convince the readers on the issue being discussed. The conclusion part usually expresses promise or offers some recommendations to the readers. In most cases, the conclusion repeats the thesis statement with slight rephrasing still intending to convey the same views put forward earlier.

For convincing the readers, the authors of such persuasive texts provide relevant evidences (facts and\or opinions) with examples, make use of logical connectives like 'Firstly',

[1] Adapted from the National Literacy Strategy Grammar for Writing p154/5

'Secondly', 'Finally', 'Because', 'Consequently', 'So', 'Therefore' etc. to structure and link the ideas within arguments.

Other persuasive devices that are often used in such texts include information dealing with statistics and numbers (for example, 'More than 80%...'), emotive words (for example, strong adjectives and adverbs like 'alarming', 'surely' etc.) and rhetorical questions like 'Are we meant to suffer like this when we have been toiling so hard?'.

Editorials, which align more closely to persuasive texts than argumentation texts are found to adhere closely to the classical definition and structure of argumentation – proposition or thesis statement followed by supports and finally the conclusion [2-4].

III. Building A Corpus Of Editorials And Op-Eds

For studying the structure of editorials, editorials are gathered for the time span 2007 – 2012, from two local English news portals from Nepal, respectively, 'The Kathmandu Post' (http://ekantipur.com/tkp/), 'Nepali Times' (http://nepalitimes.com) and similarly op-eds from three international English news portals, namely, 'BBC' (http://bbc.co.uk), 'Aljazeera' (http://aljazeera.com) and 'The Guardian' (http://guardian.com).

The study shows that the editorials and op-eds from all of the news portals exhibit a more or less similar structure adhering to persuasive texts with the following characteristics:

- Every paragraph has a thesis statement or introduction of an issue, which is elaborated or provided supports further in the paragraph thus confirming that they do follow the structure identified above.
- In terms of discourse, each paragraph represents a separate view point necessarily consolidating the views or providing supports to the topic of the editorial or overall discourse.
- The supporting statements in the paragraph are linked to each other via rhetorical relations and signaled by the logical connectives or discourse cues.
- The overall orientation of the supporting statements (Positive or Negative) can be analyzed by evaluating the opinion words or phrases occurring in the individual statements.
- The strength or the intensity of the opinions expressed in statements can be determined by evaluating the intensifiers or pre-modifiers coming in front of opinions and similarly by judging the presence of report and modal verbs that signal the commitment or intent level of the opinions.

The above findings pinpoint that the development of suitable linguistic resources can prove vital for providing at least partial solutions to the given task. In Table I, the statistics of the downloaded editorials and op-eds are presented.

TABLE I. Download Statistics of Editorials and op-eds

Source	Downloads (texts files)
The Kathmandu Post	1718
Nepali Times	211
BBC	853
Aljazeera	1830
The Guardian	6191

IV. Devising A Semantic Tagset For Annotating The Corpus

There have been growing efforts in developing annotated resources so that they can be useful in acquiring annotated patterns using statistical or machine learning approaches and ultimately aid in the automatic identification, extraction and analysis of opinions, emotions and sentiments in texts. Some of such works on text annotation, among many others, include [5-8].These works are primarily focused on annotating opinions or appraisal units (attitude, engagement and graduation) in texts, which share similar notions with the Appraisal Framework developed by [9]. Other works on annotating texts include [10, 11] etc. which deal with text annotation in the discourse level employing discourse connectives and discourse relations. However, despite these efforts, the development of a suitable annotation scheme for corpus annotation from the perspective of opinion and argumentation analysis in opinionated texts seem to be clearly missing. While the existing annotation schemes and guidelines may be sufficient for annotating appraisal units, discourse units and even possibly some rhetorical relations, for analyzing the argumentation structure, it is necessary to determine the type of supports with respect to a statement (either "For" or "Against") and the commitment or intent levels of the opinions and the overall persuasion effects in opinionated texts. This then requires for this research work to make some additional provisions in the annotation scheme which are as follows:

- Introduction of some metadata of the source text like date and source of publication useful for source attribution in opinionated texts.
- Parameters for identifying arguments and for determining the orientation of their supports.
- Attributes for determining the strength of opinions and arguments or commitment level expressed in the form of different modal and report verbs.
- Other forms of expressions indicating persuasion effect of opinions and arguments (mostly involving words or phrases consisting of one or more adjectives, adverbs, intensifiers, pre-modifiers in combination or in isolation).

With the above issues in consideration and after manually analyzing selected opinionated texts from the corpus, a semantic tagset was developed specifically designed for the annotation of the opinionated texts, a sample of the tagset and brief explanation of the tags is provided in Table II below:

TABLE II. SEMANTIC TAGSET

Parameters	Possible values/Explanations
Topic	The title or topic of the opinionated text
Gist	The summary or abstract of the opinionated text. Usually, this is provided in the form of one or more sentences at the beginning of each text.
Author	The name of the author if available. Generally in editorials, the name of the author is not provided but in case of op-eds, usually, the names of the author(s) are mentioned.
URL	The uniform resource locater or the web link to the opinionated text.
Date	The date of publication of the opinionated text.
Source	The source or the news portal from where the opinionated text is taken from.
argument_id	The argument's identity number. For simplicity, in this annotation scheme, each paragraph is regarded as an argument. This is because in argumentative text, the basic rule is that a paragraph generally sticks to a particular idea with several supporting/refuting evidence to the given idea. The numbering of the argument starts from 0 and this increases globally in the whole text as the paragraphs advance from top to bottom.
statement_id	The statement/sentence number within an argument or paragraph. Each sentence is considered to be a statement. The numbering of the statement starts from 0. The numbering of the statement is relative to each paragraph.
statement_type	Can be either a "thesis statement" or "support statement" but not both. Usually, a thesis statement puts forward a claim or a belief and the support statement supports or refutes the claim.
support_type	A statement or sentence can take either of the three values – "For" or "Against" or "Neutral". If the supporting statement supports the claim, it is said to be providing a positive support or "For" and if the supporting statement refutes the claim, it is said to be providing a negative support or "Against". Similarly, if the supporting statement does not support or refute the claim, it is said to be neutral, "Neutral" with respect to the claim.
exp_type	A statement or sentence as an expression can take either of the three values – "Opinion", or "Fact" or "Undefined". A statement is tagged as an opinion if it represents a view, emotion, judgment etc. Similarly, a statement is tagged as fact if it expresses some factual information. If a statement cannot be tagged as an "Opinion" or a "Fact", it is tagged as "Undefined". Often, there may be situations whereby a portion of a statement represents a fact while the other portion is an opinion. However, currently we handle just statements with either factual or opinionated expressions but not both.
fact_authority	If a statement or sentence has been tagged as "Fact", the attribute "fact_authority" can take either "Yes" or "Est." depending upon whether the fact has an authority to confirm about its authenticity or that it is an established fact. For well-established facts like "The earth is round" or "The sun rises from the east and sets in the west", the attribute "fact_authority" takes the value "Est.", meaning "Established".
opinion_orientation	If a statement or sentence has been tagged as "Opinion", the attribute "opinion_orientation" can take either of the three values –"Positive", "Negative" or "Neutral". There can be one or multiple opinion terms of different polarity or orientation in a statement but the statement has to be tagged taking into consideration the overall effect in terms of opinion orientation. If the statement does not bear any particular opinion orientation, i.e., either "Positive" or "Negative", it is tagged as "Neutral".
opinion_strength	This attribute tags a statement or sentence for the overall opinion strength across seven extended scale parameters - "Lowest" or "Lower" or "Low" or "Average" or "High" or "Higher" or "Highest". The general basic strength categories are however, "Low", "Average" and "High" with the other four grades resulting when one or more intensifiers or pre-modifiers come in front of the three basic strength categories. A statement can have multiple opinion terms of varying strengths but the overall opinion strength has to be considered.
persuasion_effect	This attribute tags a statement or sentence with one of the values – "Yes" or "No". If the sentence or statement has an overall persuasion effect or is of convincing nature, the attribute "persuasion_effect" takes the value "Yes", otherwise, it takes a "No" value.
Conditional	This attribute tags a sentence or statement with one of the values "Yes" or "No". If the statement is of conditional nature, the attribute "conditional" takes the value "Yes", otherwise, it takes a "No" value.
commitment_level	This attribute tags a statement or sentence with one of the values – "Low", "Average" or "High". The major decision to tag the sentences with one of the above values is determined by the presence of different modal and\or reporting verbs of varying commitment or intent levels.

rhetorical_relation_type	This attribute tags the support statement or sentence with one of the following values – "Exemplification", "Contrast", "Justification", "Elaboration", "Paraphrase", "Cause-Effect", "Result", "Explanation", "Reinforcement" and "Conditional". The tagging for the given attribute is based on explicit or implicit discourse markers or connectives present in the support statement with respect to the thesis statement or in between the preceding or following support statements with respect to the current support statement.

V. Development Of Linguistic Resources

For annotating the editorials and op-ed texts from the corpus with opinion and argument attributes as mentioned in the semantic tagset, some linguistic resources were developed within this research work, which is described in the following sections.

A. Sentiment/Polarity Lexicon

Sentiment/Polarity Lexicon represents as a valuable resource for determining the orientation or polarity of opinions in opinionated texts, particularly in the word, phrase and sentence levels. A few of such lexicons already exist for the English language, for example, the opinion lexicon developed by [12,13], subjectivity clues developed by [14,15], SentiWordNet developed by [16]. However, it should be noted that these lexicons in themselves do not serve as exhaustive lists as new opinion terms keep on coming up quite often over time with new domains. For the given task of analyzing opinions and arguments in opinionated tasks, the opinion lexicon for English by [12] is taken as a baseline resource, which consists of 2041 positive terms and 4818 negative terms. This lexicon was found to be quite useful for the given work and effectively helps in determining opinion bearing words and their orientation or polarity but it was found that the resource quickly breaks down with terms from the socio-political domain. Even the frequent terms like 'treaty', 'pact', 'truce', 'agitation', 'mutiny', 'salvage', 'consensus', 'epidemics', 'brotherhood', 'bandh' etc. in the socio-political domain seem to be missing in the opinion lexicon. This motivated the author to develop a separate sentiment polarity lexicon comprising of prototypically positive and negative terms, specifically from the corpus. The lexicon development started with a small collection of 29 positive terms and 73 negative terms from the corpus. These terms were collected by a manual analysis of some random texts from the corpus. Further, consulting the online and available electronic resources like dictionaries, thesaurai and the WordNet, the list of terms was extended by adding some synonyms, inflected and derivational forms of the words. A sample of the developed Sentiment/Polarity Lexicon is presented in Table III. Such a collection allows having a rich lexicon of wider coverage comprising of both domain-specific terms from the corpus and domain independent terms from online resources. Currently, the Sentiment/Polarity terms contains about 300 positive terms and 800 negative terms. The given task of opinion and argument analysis in opinionated texts involves analyzing the opinions in the lexical and phrase levels first and then assigning an opinion label – Positive or Negative or Neutral to each statement/sentence. To illustrate the use of the Sentiment/Polarity Lexicon in the process of opinion analysis in the lower levels (lexical and phrase) and the assignment of opinion label in the sentence level, an excerpt of the real text from the corpus and its corresponding opinion analysis is presented in Fig. 1.

TABLE III. Sample of the Sentiment/Polarity Lexicon

Positive	Negative
right : proper, correct, ok, okay ***reform***: reforms, reformed ***democracy***: democratic, democratized ***contribute***: contributed, contribution ***hope***: hopeful, hoping ***thank***: grateful, gratitude, thankful ***respect***: honor, dignity, diginified, respectful ***integrate***: unite, unity, united, integrated, integration, merge ***salve***: salvage, save ***glory***: glorious, famous	***sack***: fire, throw ***insubordinate***: insubordination ***defy***: disobey, defiance ***unilateral***: unilaterally ***withdraw***: withdrew, withdrawal ***hate***: hated, hatred ***damage***: damaging, damaged ***contradict:*** contradiction, contradicting ***insurgent***: insurgency ***refuse:*** refusal, denial

For ease of illustration, the text is segmented in the sentence level and also analyzed for opinions in the lexical and phrase levels. While opinion phrases are annotated in XML like tagging notation, the opinion words/expressions have been underlined.

TITLE@Maoists' double standard
DATE@2007 May 05
#URL@http://ekantipur.com/the-kathmandu-post/2007/05/05/editorial/maoists-double-standard/108572.html

1. A report of the UN Office of the High Commissioner for Human Rights in Nepal (OHCHR-Nepal), issued last week, manifests the **<neg>**glaring facts**</neg>** about the CPN-Maoist. **{Overall orientation: Negative}**
2. In the report the OHCHR-Nepal has starkly said that the Maoist cadres **<neg>**aren't complying**</neg>** with their party's commitments and **<neg>**are not respecting**</neg>** the rights of the Internally Displaced Persons (IDPs) to voluntarily and safely return home.**{Overall orientation: Negative}**

Fig. 1. Excerpt of the analyzed text from the corpus for opinion orientation

B. Intensifier/Pre-modifier Lexicon

For the task of analyzing the opinions and arguments in opinionated texts, besides determining the subjectivity (whether a given expression is an opinion or not) and detection of the orientation or polarity of opinions, it is also necessary to assess the strength or degree or intensity of opinions. Adjectives and adverbs have a significant role in the determination of the strength or degree of opinions as they necessarily change the intensity or degree of opinions being expressed [17-20]. Although, there can be finer grades of any opinion, we have limited the grading to seven broad scales – "Lowest", "Lower", "Low", "Average", "High", "Higher" and "Highest" for our task. This correspond to a scale within the range -3 to 3, where the mapping of the degrees to numeric values are as follows:

Lowest = -3; Lower = -2; Low =-1; Average = 0, High=1;
Higher=2, Highest=3

The mapping above is partly guided by the three degrees of adjectives in English, viz., positive, comparative and superlative.In our case, positive degree refers to "Low", comparative degree to "Average" and superlative to "High". These three scales have been considered as our base strength categories. The remaining four scales "Lower" and "Lowest" and "Higher" and "Highest", respectively on the "Low" and "High" sides are produced as a result of the possible occurrence of intensifiers and pre-modifiers in front of the three major degrees of adjectives – "Low", "Average" and "High". Below, a few examples of the three degrees of adjectives from the corpus have been provided:

high, low, good, bad, few, wealthy, powerful, successful:	**positive degree** ("**Low**")
higher, lower, better, worse, fewer, wealthier, more powerful, more successful:	**comparative degree** ("**Average**")
highest, lowest, best, worst, fewest, wealthiest, most powerful, most successful:	**superlative degree** ("**High**")

In addition to adjectives, the given work also considers intensifiers and pre-modifiers for the determination of the different degrees of strength of opinions. Intensifiers are essentially adverbs which are reported to have three different functions – emphasis, amplification and downtoning. Pre-modifiers, on the other hand, come in front of adverbs and adjectives. Both intensifiers and pre-modifiers play a role in conveying a greater and/or lesser emphasis to do something. A sample of the intensifier lexicon is presented in Table IV below:

TABLE IV. SAMPLE OF THE INTENSIFIER LEXICON

Type	***Value***	***Occurrences from the Corpus***
Emphasizer	**Really**: truly, genuinely, actually **Simply**: merely, just, only, plainly **Literally** **For sure**: surely, certainly, sure, for certain, sure enough, undoubtedly **Of course**: naturally	This is **really** a good idea. I **simply** cannot say. I would **literally** trust his judgments over mine. All we can say **for sure** at this point is … There were many tactical and strategic compromises along the way, **of course**.
Amplifiers	**Completely:** all, altogether, entirely, totally, whole, wholly. **Absolutely:** totally, definitely, without question, perfectly, utterly. **Heartily:** cordially, warmly, with gusto and without reservation.	Men and women are **completely** equal in value and dignity. I just told them that we should be **absolutely** quiet. **Heartily** approve of socialism.
Downtoners	**Kind of:** sort of, kinda, rather, to some extent, almost, all but **Mildly:** gently	The opponents were **kind of** satisfied with the answers of the Prime Minister. The Prime Minister **mildly** protested the proposal.

Below, the role of each category of intensifiers in terms of modifying the strength of opinions in example texts from the corpus is discussed:

"The loss of the Corby bi-election is a **really** significant watershed".

The intensifier "really" emphasizes the adjective "significant", thus increasing its intensity or degree to one level further up. In this respect, since the adjective "significant" represents the positive or "Low" degree, the intensifier "really" modifies the intensity of strength of the adjective to "Average".

"The electoral Commission was **absolutely** right to announce a review of the debacle".

Similarly, the intensifier "absolutely" amplifies the adverb "right", thus increasing its intensity or degree to the highest level. In this respect, the intensifier "absolutely" modifies the intensity of the strength of the adverb to "Highest".

"Admittedly, this sounds **rather** disconcerting."

Likewise, the intensifier "rather" downtones the adverb "disconcerting" to one level down, thus modifying the intensity of the strength of the adverb to "Lower". Similarly, in Table V, a sample of the pre-modifiers lexicon is presented and the contribution of the pre-modifiers to the overall strengths of the opinion expressions is shown.

TABLE V. SAMPLE OF THE PRE-MODIFIERS LEXICON

Adverb/Adjective (Initial strength)	**Pre-modifier**	**Modified strength**
Fast (Low)	Very	Very fast (High)
Careful (Low)	Lot more	Lot more careful (High)
Better (Average) Serious (Low)	Much	Much better (High)
		Much much better (Higher)
		Much more serious (Higher)
Good (Low)	Somewhat	Somewhat good (Average)
	Quite	Quite good (Average)

C. Report and Modal Verbs Lexicon

For the task of determining the strength of opinions and arguments in opinionated texts, it is also necessary to analyze the intent or commitment level of the statement under consideration with respect to some thesis statement. One way of doing this is by looking at the choice of report or modal verbs used in the respective statements.

The higher the degree of assertiveness a modal/reporting verb represents, the stronger the commitment or intent level of the statement would be. In Table VI, a sample of the modal verb lexicon is presented and the role of modal verbs in commitment or intent level determination is illustrated.

TABLE VI. SAMPLE OF THE MODAL VERBS LEXICON

Type	**Verb**	**Strength effects**
Ability/Possibility	Can	Average
Ability/Possibility	Could	Low
Permission	May	Average
Permission	Might	Low
Advice/Recommendation/Suggestion	Should	Average
Necessity/Obligation	Must, Have to	High

Similarly, in Table VII, we present a sample of the Report Verb Lexicon.

TABLE VII. SAMPLE OF THE REPORT VERBS LEXICON

Type	***Low***	***Average***	***High***
Agreement	admits, concedes	accepts, acknowledges, agrees	Agreement
Argument and persuasion	Apologizes	assures, encourages, interprets, justifies, reasons	Argument and persuasion
Believing	guesses, hopes, imagines	believes, claims, declares, expresses	Believing
Disagreement and questioning	doubts, questions	challenges, debates, disagrees, questions	Disagreement and questioning
Presentation	Confuses	comments, defines, reports, states	Presentation
Suggestion	alleges, intimates, speculates	advises, advocates, posits, suggests	recommends, urges

Source:[http://www.adelaide.edu.au/writingcentre/learning_guides/learningGuide_reportingVerbs.pdf]

To illustrate the use of the Intensifiers and Pre-modifiers Lexicon as well as the Report and Modal Verbs Lexicon for determining the commitment or intent level of the statements, an excerpt of real text from the corpus and its corresponding analysis is presented in Fig.2. below:

Along with the laundry list of domestic grievances **<commitment_level="Average">**expressed **</commitment_level>**by Egyptian protesters **<commitment_level="High">**calling**</commitment_level>** for an end to the regime of Hosni Mubarak, the popular perception of Egypt's foreign policy has also been a focal point of the demonstrations.**{Overall commitment level: "High"}**

Fig. 2. Excerpt of the analyzed text from the corpus for commitment level

For the determination of the overall commitment level and the opinion strength in the sentence level, the highest values available within the sentence for each of these two attributes has been taken.

D. Discourse Markers and Rhetorical Relations Lexicon

For analyzing the opinions and arguments in the sentence and higher levels, the rhetorical or discourse or coherence relations needs to be determined. These relations are crucial in establishing relationships between passages of text.

Discourse markers can serve as effective sign posts to signal the presence of discourse or coherence or rhetorical relations in any discourse [21,22]. In Table VIII, a sample of the Discourse Markers and Rhetorical Relations Lexicon is presented.

TABLE VIII. SAMPLE OF THE RHETORICAL RELATIONS AND DISCOURSE MARKERS LEXICON

Rhetorical relations	Discourse Markers
Elaboration	after, before, first, all the while, in the past, …
Result	briefly, hence, overall, thus, in brief, to end,…
Reinforcement	again, also, too, in addition, above all, most of all, …
Contrast	against, instead, rather, still, versus, yet, even so,…
Cause – Effect	hence, since, therefore, thus, whenever, as a result, …
Exemplification	indeed, namely, for example, in effect, such as, …
Conditional	else, if, otherwise, unless, until, while, as long as, …

Source:[http://learning.londonmet.ac.uk/TLTC/connorj/WritingGroups/Writing/5%20discourse%20markers-signposts.pdf]

To illustrate the use of the Discourse Marker and Rhetorical Relations Lexicon in analyzing the discourse or coherence or rhetorical relations between supporting statements in texts, an excerpt of real text from the corpus and its corresponding analysis is presented in Fig.3. below. The text fragments having the discourse markers have been underlined in the figure.

TITLE@In praise of ... Jimmy Carter
DATE@2008 Apr 18
#URL@http://www.theguardian.com/commentisfree/2008/apr/18/usa
<**Rhetorical_relation="Exemplification"**>Like the Kennedy Library in Boston, where Gordon Brown makes the main foreign policy speech of his US visit today, most American presidential libraries are monuments to the past.</**Rhetorical_relation**>
<**Rhetorical_relation="Contrast"**>The Carter Centre, near Atlanta, is totally different.</**Rhetorical_relation**>
<**Rhetorical_relation="Exemplification"**>Like its begetter, Jimmy Carter, it is focused on the future.</**Rhetorical_relation**>

Fig. 3. Excerpt of the analyzed text from the corpus for rhetorical relations

VI. DEVELOPMENT OF AN AUTOMATIC ANNOTATION TOOL AND EVALUATION OF PERFORMANCE

Based on the linguistic resources described in the previous section, an automatic annotation tool has been developed, which segments the text into paragraphs and sentences, then annotates the text for opinions and arguments with the attributes of the semantic tagset. For the evaluation of the performance of the annotation tool, 500 texts have been randomly taken from the 10,000 automatically annotated texts by the tool. The accuracy of the performance of the tool was evaluated manually in terms of annotations by the machine compared to what a human would have annotated for the same. Since the annotation tool highly relies on the linguistic resources developed in terms of annotation, a comparative analysis of the use of the baseline linguistic resource (opinion lexicon by [12]) versus our extended linguistic resource (sentiment/polarity lexicon by [12]augmented with domain specific opinion terms and patterns) for the same 200 texts mentioned above was carried out. The accuracy of the performance of the automatic tagger application in terms of tagging was calculated as follows:

$$Accuracy = {tag}/{T} \quad \text{.....(1)}$$

Where T = Total number of tagged sentences
tag = Total number of correctly tagged sentences

The accuracy scores for the different annotation tasks are presented in Table IX below:

TABLE IX. ACCURACY SCORES FOR THE DIFFERENT TAGGING TASKS

S.No.	Annotation task	Accuracy (%)
1	Opinion orientation	61.5%
2	Opinion strength	63.75%
3	Commitment or intent level	72.5%
4	Rhetorical relations	47.5%

Similarly, in Table X, the accuracies of the annotation tool for the attribute 'opinion_orientation' using the baseline resource and our extended linguistic resource are presented.

TABLE X. ACCURACY SCORES FOR BASELINE AND EXTENDED LINGUISTIC RESOURCES

S.No.	Annotation Task (Opinion Orientation) versus Linguistic Resources	Accuracy (%)
1	Baseline Linguistic Resource	55%
2	Extended Linguistic Resource	68%

The accuracy scores in Table IX show that the annotation tasks have achieved reasonably good results. The scores for each of these individual tasks are expected to further improve as the linguistic resources are further enhanced in terms of coverage and size. The task currently performing the least is the determining the rhetorical relations. This is partly because implicit discourse markers in texts, which also potentially act as signposts for denoting the presence of rhetorical relations in between statements, have not been considered at the moment. The performance of the tool for this particular task is expected to further improve as some special tailored rules designed to address such situations are developed.

Similarly, the accuracy scores in Table X show that the performance of the tool using the extended linguistic resource is better than using the baseline linguistic resource. This is understandable as the extended linguistic resource has a rich collection of domain specific terms from the corpus in addition to the opinionated terms from the baseline linguistic resource. The accuracy scores of the tool using the extended linguistic resources is expected to improve further as more of such domain specific terms and patterns are gathered.

VII. CONCLUSION AND FUTURE WORKS

The paper presented on the ongoing efforts towards developing linguistic resources for automatic annotation and consequently analysis of opinions and arguments in editorials

and op-eds. An automatic annotation tool developed for this purpose was reported to be performing with reasonably good accuracies. Currently, the annotation tool basically relies heavily on the linguistic resources and some contextual rules to annotate the texts for opinions and arguments. In due course of time, some machine learning capabilities are being planned to incorporate to the tool so that the same task can be handled more accurately and in a larger scale. There are also plans to work on building a synthesis of opinions and arguments on a particular topic from multiple editorial sources. Such a synthesis helps to get more or less a true picture of the events and at the same time also potentially reveal the inherent biases and prejudices. At the moment, works are underway for developing a framework for creating such a synthesis.

ACKNOWLEDGMENT

I would like to extend my sincere thanks to the University Grants Commission, Nepal for supporting this work. My sincere thanks also go to my Research students, Mr. Chandan Prasad Gupta and Mr. Rohit Man Amatya at the Information and Language Processing Research Lab, Department of Computer Science and Engineering, Kathmandu University for their technical help for conducting this Research work. I would similarly like to thank my PhD advisors, Prof. Patrick Saint-Dizier from IRIT Labs, Toulouse, France and Prof. Patrick A.V. Hall, Kathmandu University for their continuous help and guidance to this work.

REFERENCES

[1] B.K. Bal, P. Saint-Dizier (2010). Towards Building Annotated Resources for Analyzing Opinions and Argumentation in News Editorials. *LREC, Malta, ELRA.*

[2] H. Stonecipher (1979). *Editorial and Persuasive Writing:Opinion Functions of the News Media.* New York: Communication Arts Books. Hastings House, Publishers.

[3] T. Van Dijk (14-17 December, 1995). Opinions and Ideologies in Editorials. *Paper Symposium of Critical Discourse Analysis. Language, social life and critical thought.* Athens.

[4] N. B. Wekesa (2012). Assessing Argumentativity in the English Medium Kenyan Newspaper Editorials from a Linguistic-Pragmatic Approach. *International Journal of Humanities and Social Science , 2* (21), 133-144.

[5] T. Wilson (2003). Annotating Opinions in World Press. *Proceedings of the SIGdial-03*, (pp. 13-22).

[6] T. Wilson (2005). Annotating Attributions and Private States. *In Proceedings of the ACL 2005 Workshop: Frontiers in Corpus Annotation II: Pie in the Sky*, (pp. 53-60).

[7] V. Stoyanov, C. Cardie, D. Litman, & J. Wiebe (2004). Evaluating an Opinion Annotation Scheme Using a New Multi-Perspective Question and Answer Corpus. (Q. S. Wiebe, Ed.) *Computing Attitude and Affect in Text: Theory and Practice* , 77-89.

[8] J. Read, D. Hope, & J. Carroll(2007). Annotating Expressions of Appraisal in English. *Proceedings of the ACL 2007 Linguistic Annotation Workshop.* Prague, Czech Republic.

[9] J. Martin, & P. R. White (2005). *The Language of Evaluation: Appraisal in English.* London: Palgrave: Macmillan.

[10] L. Carlson, D. Marcu & M. Okurowski (2001). Building a Discourse-tagged Corpus in the Framework of Rhetorical Structure Theory. *In Proceedings of the Second Sigdial Workshop on Discourse and Dialogue. 16*, pp. 1-10. Aaolborg, Denmark: Annual Meeting of the ACL, Association for Computational Linguistics, Morristown, NJ.

[11] M.Taboada, & J. Renkema (2008). *http://www.sfu.ca/rst/06tools/discourse_relations_corpus.html.* Retrieved August 18, 2013, from http://www.sfu.ca.

[12] M. Hu & B. Liu (2004). Mining and Summarizing Customer Reviews. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Aug 22-25, 2004.* Seattle, Washington, USA.

[13] B. Liu, M. Hu & J. Cheng (2005). Opinion Observer: Analyzing and Comparing Opinions on the Web. *Proceedings of the 14'th International World Wide Web Conference (WWW-2005), May 10-14, 2005.* Chiba, Japan.

[14] E. Riloff, J. Wiebe, & T. Wilson(2003). Learning Subjective Nouns Using Extraction Pattern Bootstrapping. *In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4 (CONLL '03). 4*, pp. 25-32. Stroudsburg, PA, USA: Association for Computational Linguistics.

[15] T. Wilson & J.M.Wiebe. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. *Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, (pp. 347-354).

[16] A. Esuli, & F. Sebastiani (2006). SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. *In Proceedings of the 5'th Conference on Language Resources and Evaluation (LREC '06)*, (pp. 417-422).

[17] V. Hatzivassiloglou & K. R. McKeown (1997). Predicting the semantic orientation of adjectives. *In Proceedings of the Eigth Conference on European Chapter of the Association for Computational Linguistics (EACL '97)* (pp. 174-181). Stroudsburg, PA, USA: Association for Computational Linguistics.

[18] V. Hatzivassiloglou & J. M. Wiebe (2000). Effects of Adjective Orientation and Gradability on Sentence Subjectivity. *In Proceedings of the 18'th Conference on Computational Linguistics (COLING '00). 1*, pp. 299-305. Stroudsburg, PA, USA: Association for Computational Linguistics.

[19] P. Chesley, B. Vincent, L. Xu, & R. Srihari (2006). Using Verbs and Adjectives to Automatically Classify Blog Sentiment. *In Proceedings of AAAI-CAAW-06, the Spring Symposia on Computational Approaches to Analyzing Weblogs.*

[20] F. Benmara, C. Cesarano, A. Picariello, D. Reforgiato, & V. Subrahmanian (2007). Sentiment Analysis: Adjectives and Adverbs are Better than Adjectives Alone. *Proceedings of the International Conference on Weblogs and Social Media (ICWSM).*

[21] D. Marcu (1998). A Surface-based Approach for Identifying Discourse Markers and Elementary Textual Units in Unrestricted Texts. In S. Manfred, L. Wanner, & E. Hovy (Ed.), *Proceedings of COLING-ACL Workshop on Discourse Relations and Discourse Markers*, (pp. 1-7). Montreal, Canada.

[22] B. Fraser (1999). What are discourse markers? *Journal of Pragmatics* (31), 931-952.

Automating the Shaping of Metadata Extracted from a Company Website with Open Source Tools

DR IR ROBERT VISEUR CETIC
Rue des Frères Wright, 29/3 6041 Charleroi, Belgium
UMONS Faculty of Engineering
Rue de Houdain, 97000 Mons,
Belgium

***Abstract* — As part of a market analysis process, the objective was to automate the task of identifying the activities and skills of a collection of enterprises, namely Belgian and French open source companies. In order to avoid manual annotation through visual analysis of the websites' content, a tool chain was developed to collect the content of websites and extract the important terms. Standard software libraries were identified, allowing to clean up HTML documents and to perform the part-of-speech tagging process used for extracting terminology. This procedure is supplemented by the extraction and the recognition of named entities. The terms extracted in the HTML pages of a company website were then merged and filtered and a circular tags cloud was generated. This presentation facilitates the identification of important terms, commonly referred to as activities and technologies supported by the company. Several changes are planned for this prototype, including, in particular, the extension to the texts in French, the association of extracted terms to the vocabulary of a classification scheme and the automatic generation of dashboards to facilitate the monitoring of the evolution of the industrial sector.**

Keywords—terminology extraction; named entities; NLP; tag cloud; market analysis

I. INTRODUCTION

As part of a market analysis process, a business directory needs to be maintained. Each entry is associated with a set of keywords to characterize the activities, technologies and software supported by the companies. These keywords are determined by the communication implemented by the company on its website. They are used to find specialized providers, to explore the market from a tag cloud and generate dashboards to compare the relative weight of technologies supported by the suppliers. This process to explore, annotate and update metadata is time-consuming. Research was therefore undertaken to accelerate and automate this task by establishing an information retrieval system, preferably based on standard tools.

This paper is organized in three parts. First, a state of the art on techniques and tools used to solve our problem was carried out. How to extract keywords from the content of a Web site will first be analyzed. This step will focus on two distinct problems; the conversion from HTML documents to raw text and the extraction of keywords to qualify business activity. A state of the art on techniques used to extract terminology and named entities will therefore be presented.

How to present the results of the extraction in order for the main topics of the company website to be understood will also be covered, followed by a presentation of results of a first implementation of a tool that extracts and formats keywords from a website. To conclude, possible improvements to this system will be discussed.

II. BACKGROUND

The metadata extraction process can be divided into three stages [7]:

1) *the conversion and the standardization of source files,*
2) *the part-of-speech tagging (POST),*
3) *the extraction of metadata.*

This process must be followed by the visualization of the extracted metadata.

B. Conversion of source files

The conversion of source files (in this case, HTML documents) is important because it determines the quality of the next step designed for part-of-speech tagging. Indeed the accuracy of the part-of-speech tagging influences the accuracy of the retrieval algorithms [22]. In practice, part-of-speech taggers often malfunction with data from the Web for various different reasons [2, 10]. Following the cleanup of HTML documents, the text entered into the tool may end up containing parasitic features, such as textual elements belonging to the menus, scripts, style sheets and footers. Additionally, the text may not meet the standards of written English (e.g. spelling errors, grammatical errors and specific writing styles) and may need to be standardized [9]. The taggers are also designed for a language (or set of languages) hence the need for prior configuration of the language of the document. Several approaches are possible for cleaning up documents.

A first approach is based on general tools for a crop (in French: "*détourage*") of the document. The term "crop" is proposed by Dutrey *et al.* by analogy with image processing. This means the separation of text and code in the context of digital structured or semi-structured documentation and/or separation within the textual content between relevant and irrelevant text [9]. Regarding HTML, Boilerpipe (refer to http://code.google.com/p/boilerpipe/) is a reference tool which enables an automatic cleanup of content pages and is distinguished by its good performance of low calculation time, ease of use and high accuracy [11].

A second approach relies on the special characteristics of the analyzed documents to extract content with better precision [17]. The use of reverse engineering tools for Web pages is possible when implementing this approach. These tools allow the content to be targeted more precisely and a first layer of semantics to be added. They may require writing extraction rules in the HTML document and may need to be distinguished by their ability to generate these rules semi-automatically [16]. However, they often use technologies such as XPath or XQuery, and can suffer from a lack of markup validity of HTML Web documents [7].

The conversion can be followed by normalizing a text, in other words, correcting spelling errors, deleting extra spaces, homogenizing punctuation, etc.

C. Part-of-speech Tagging

The part-of-speech tagging is a process of combining the words in a text and their grammatical function (e.g. noun, verb, etc.) based on lexical and contextual information [8]. Five criteria are used to select a part-of-speech software program, such as product support, license, available languages, accuracy and processing speed [1, 10, 20]. POST tools are widely available for English, but support for other languages is often poor. Various studies can assist in choosing a POST tool [9, 13, 20, 21].

The creation of a tagger for a given language requires detailed knowledge of the language and an important preliminary annotation. Unsupervised part-of-speech taggers could help overcome these constraints [3].

D. Extraction of Metadata

The extraction of metadata involves terminology extraction techniques which operate by extracting collocations, in other words, for example, word-pairs or word-triplets. This step is the result of the part-of-speech tagging. The extractor retains collocations such as Noun-Noun, Noun-Adjective, Noun-Preposition-Noun, etc. These collocations should then be filtered. According to Zhang *et al.*, the performance of filters depends on the type of document source (specialized or more general) [22]. The authors do not recommend the simple filtering of low-frequency words and encourage testing other related filters, for example, measures made on taken on sets of documents. Some systems only retain word-pairs and word-triplets. However, single terms should not be overlooked as they can be, in some areas, significant (e.g. gene names).

These terms may be extended by named entities via a specific extraction process. The notion of named entity refers to a unique and concrete entity, belonging to a specific domain [14]. In practice, it covers proper nouns, times and amounts. The Message Understanding Conferences (MUC) lecture series proposed a categorization with three categories and seven subcategories: Named Entities / ENAMEX (organization, location and person), Temporal Expressions / TIMEX (date and time) and Number Expressions / NUMEX (money and percent). Finer categorizations also exist. Sekine and Nobata offer 200 categories [18]. Systems exceeding 2000 categories also exist. They are used for the implementation of semantic labeling [6]. The problem of language support also arises with regard to the tools used for named entity recognition.

E. Visualization of Metadata

The principle of tag cloud can be used for the visualization of extracted metadata. The tag clouds are suitable for the exploration of content and the research of resources associated with wider research [5]. Lohmann, Ziegler and Tetzlaff studied the impact of the choice of format for tag clouds [12]. This study compares the presentation of tags in several forms: as a list arranged in alphabetical order (without changing the visual properties), as a list arranged in alphabetical order (with a change of visual properties), as a circular cloud (with the most popular tags in the center) and as a clustered cloud (tags belonging to the same theme put together). Searching for keywords in an alphabetical list is more efficient than searching for them in alphabetically arranged tags cloud. However, the latter is most suitable for identifying popular tags. Searching for popular tags in a circular cloud is easier as is searching within a clustered tag cloud for theme-specific tags.

III. Implementation

The tests were carried out on a set of websites from Belgian and French companies specializing in free and open source software. As regards the selection of tools, the use of free and open source software was the preferred choice. See figure 1 for the implementation scheme.

Wget, the free software, was used to recover the website content and store it locally (refer to http://www.gnu.org/software/wget/). By following the hyperlinks, this software allows recursive crawling. It also stores the website locally and retains its original structure. It is therefore easy to refine treatments to be carried out on the original content by using the local copy. The conversion of HTML documents into plain text is supported by Boilerpipe. This software also cleans up documents without requiring a specific configuration for each website, and retains only the useful content of the HTML document.

The tools for the part-of-speech tagging and the extraction of named entities are common in English but less common in French. Therefore, there was initially access to websites in English. However, it was noticed that websites in English could contain pages (or fragments of pages) written in other languages. For example, a site that was recovered stored different language versions in containers, in other words, <DIV> tags in HTML, of which the visibility was selected according to the desired language. A language test was introduced before the part-of-speech tagging process. The language is detected by the Java language-detection library (refer to http://code.google.com/p/language-detection/). The contents are filtered by threshold on the score of the first detected language. According to the selection criteria presented in the state of the art (support, license, available languages, accuracy and processing speed), the software OpenNLP was used (refer to http://opennlp.sourceforge.net). OpenNLP is a project supported and maintained by the Apache Software Foundation. It is published under the Apache License, a permissive free license, facilitating integration into development covered by different licenses (it is noted, however, that there is an inconsistency between the Apache v2 license and the GPL v2 license). Multiple languages are available, including English, Spanish and Dutch. OpenNLP

boasts a good reputation for accuracy and processing speed (e.g.: [4], [20] and [21]).

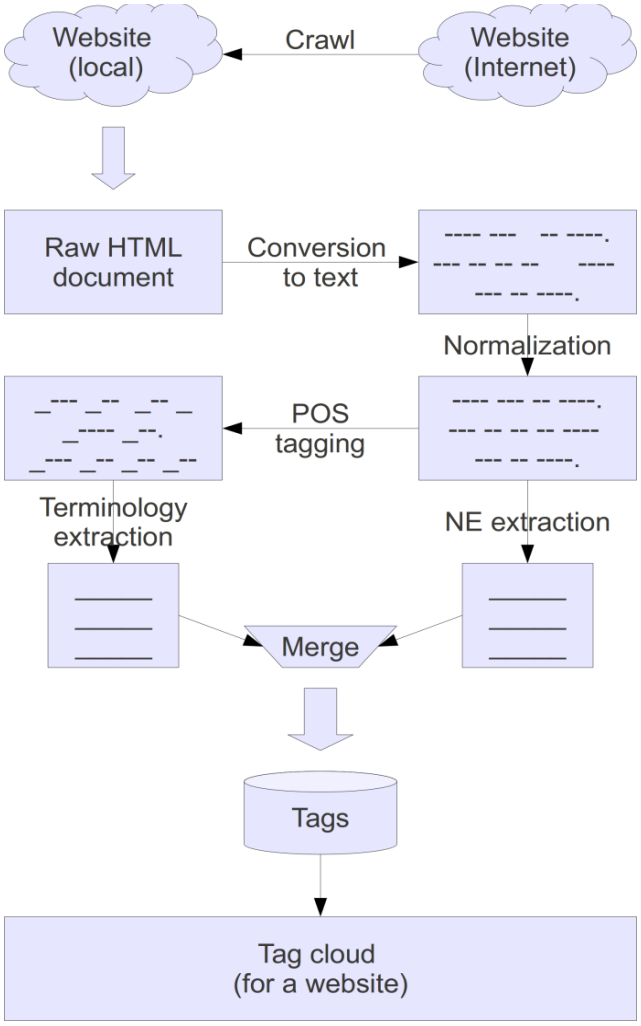


Fig. 1. Implementation scheme.

The extraction works by selecting and filtering collocations such as Noun-Noun, Noun-Adjective, Noun-Preposition-Noun, etc. Single terms were chosen if they corresponded to a proper noun, which include, typically, company names, software names, standards names, etc.

First, a minimum frequency was chosen for filtering the extracted terms. This approach remains common, yet it is criticized since it reduces the recall [22]. While the recall is not a criticism in itself and can be improved at a later stage, the principle objective is to quickly view the most common words. OpenNLP also provides the functions needed for the extraction of named entities in English, Spanish and Dutch. It was used to extract the person named entities.

Two types of formatting, for the retained terms were implemented after the filtering stage. This demonstrated the principle of the tag cloud. The first presentation involves putting the tags in a tag cloud, with an alphabetical list and highlighting the most important terms (see Figure 2).

The second presentation classifies the terms from most the important to the least important, then formats them in a circular tags cloud (see Figure 3). The circular tags cloud seems well suited to quickly view the people, themes and activities that are important for a company.

IV. Conclusion

This first prototype has validated the principle of the automated annotation of a company directory based on the content of Web sites. It also instigates interesting perspectives regarding the development of sectorial market analysis activities, the basis of this project.

This project has, however, highlighted the difficulty of having sustainable and freely available tools for part-of-speech tagging, but particularly for the extraction of named entities, as in the case of texts in French.

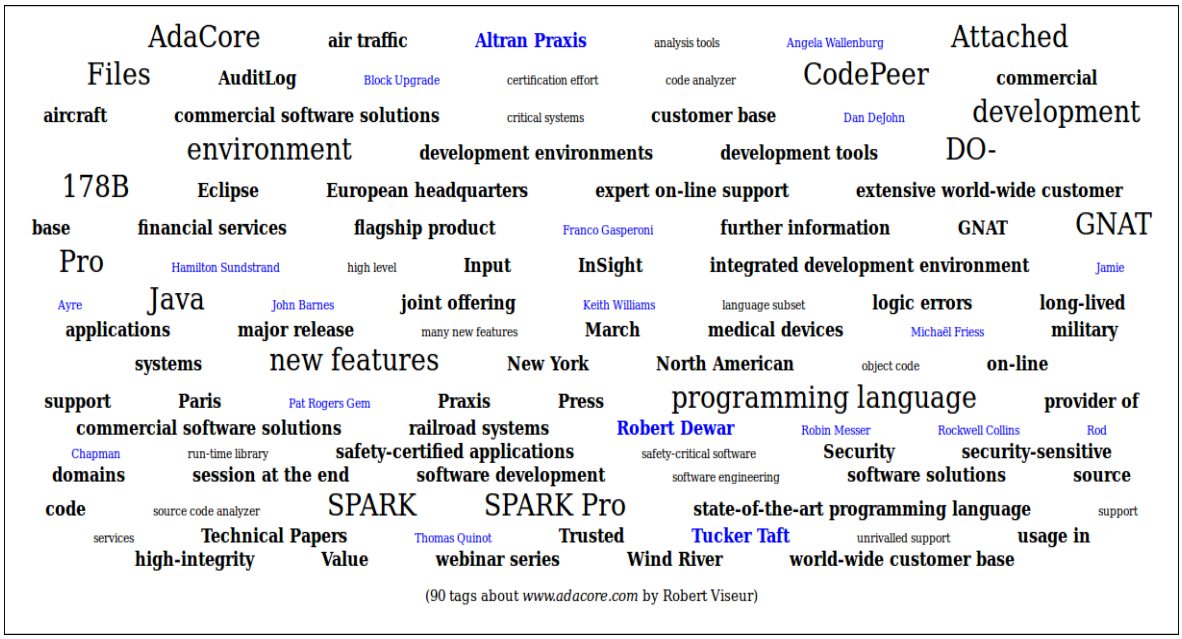


Fig. 2. Classic tag cloud.

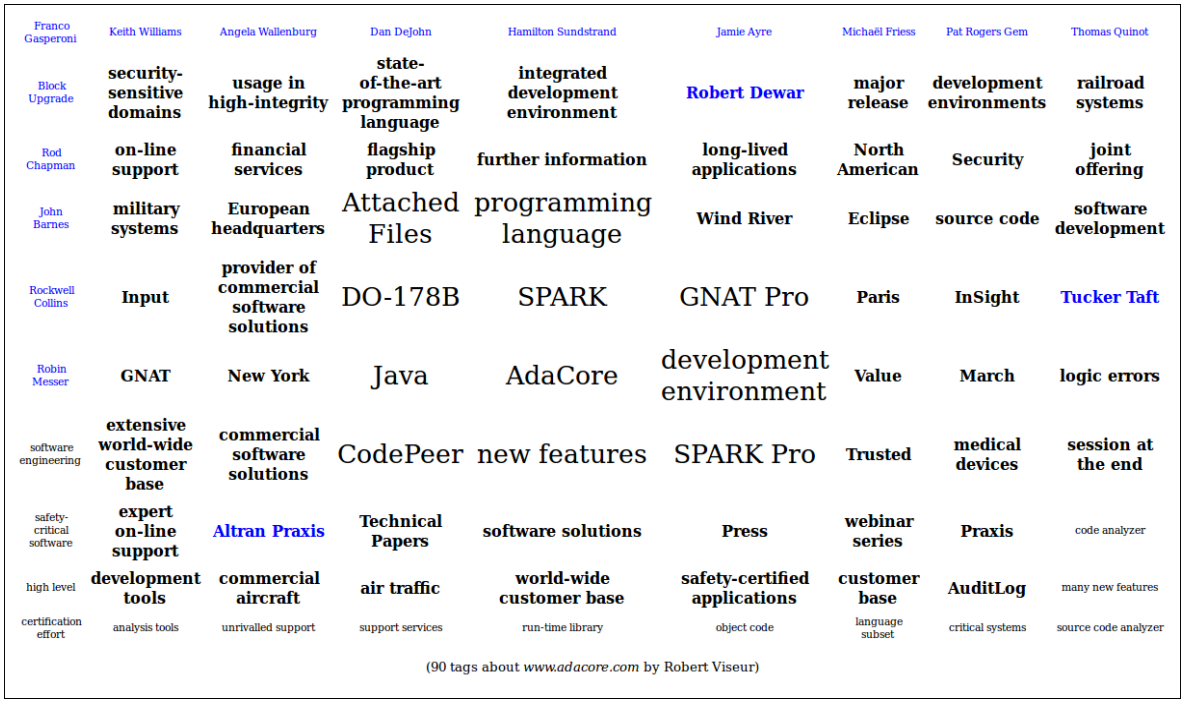


Fig. 3. Circular tag cloud.

V. FUTURE WORKS

Only the English language is currently taken into account. Considering the French language is a must due to the high number of French companies in the industrial sector that shall be studied. Adding a French model to OpenNLP is possible but requires specific skills. Using other part-of-speech software is also possible. Unsupos (refer to http://wortschatz.uni-leipzig.de/~cbiemann/software/unsupos.html) is a possibility in this case since it supports English, French, Dutch, German and Italian, as is Stanford Log-linear Part-of-Speech Tagger (refer to http://nlp.stanford.edu/software/tagger.shtml) which supports the French language in particular (since January 2012) as well as English and German [3, 19]. For the named entity extraction and recognition, a first evaluation of TagEN software was performed on a real corpus (French web content that was previously cleaned), with encouraging results for dates, locations and people, but unsatisfactory for organizations names (low recall and partial extraction for multi-word expressions).

As with folksonomies, terms extracted from a set of Web sites are subject to change depending on the communication policy of a company and the person in charge. This approach lacks controlled vocabulary. The system could therefore benefit from the establishment of a link between the extracted terms and vocabulary from a taxonomy (see [6] and [15]). Using a thesaurus (a list of controlled terms enriched by pre-defined associative relationships) or an ontology (a descriptive knowledge model based on concepts with types, properties and relations) instead of a taxonomy (a list of controlled terms organized hierarchically) would potentially lead to a more in-depth analysis of the possibilities of automatic generation of dashboards (e.g. dashboards by type of software). This may require the creation of a classification scheme, possibly using already existing tools (e.g. dictionaries, Wikipedia/DBpedia, etc.).

The ability to automatically extract common terms may allow evolutions in technology adoption to be investigated and developed. Pirolli offers this type of development for tag clouds [14]. The emergence of new tags in folksonomy can in fact show a phenomenon of craze or disinterest.

We do not use tags from folksonomy but terms extracted from websites' content (i.e. outcome of corporate communication), which can lead to the same kind of phenomenon. The terms monitored over time should allow the commercial life cycle of a technology in an industrial sector to be visualized.

REFERENCES

[1] J. Asmussen, "Survey of POS taggers", DK-CLARIN WP 2.1 Technical Report, Final version of August 19, 2011.

[2] B. Baroni, F. Chantree, A. Kilgarriff, and S. Sharoff, "CleanEval: a competition for cleaning webpages", Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), 2008.

[3] C. Biemann, "Unsupervised Part-of-Speech Tagging Employing Efficient Graph Clustering", Proceedings of the COLING/ACL-06 Student Research Workshop 2006, Sydney, Australia.

[4] F. Boudin and N. Hernandez, "Détection et correction automatique d'erreurs d'annotation morpho-syntaxique du French TreeBank", Actes de la conférence conjointe JEP-TALN-RECITAL 2012, Vol. 2, pp. 281–291.

[5] M. Cardew-Hall and J. Sinclair, "The folksonomy tag cloud: Is it useful", Journal of Information Science, vol. 34, no. 1, 2008, pp. 14-29.

[6] E. Charton, M. Gagnon, and B. Ozell, "Extension d'un système d'étiquetage d'entités nommées en étiqueteur sémantique", TALN 2010, Montréal, 19–23 juillet 2010.

[7] S. Chen, D. Hong, and V. Y. Shen, "An experimental study on validation problems with existing html webpages". In International Conference on Internet Computing ICOMP 2005, pp. 373-379.

[8] M. Dieye, M.R. Doulache, M. Floussi, J. Chabalier, I. Mougenot, and M. Roche, "Construction d'un dictionnaire multilingue de biodiversité à partir de dires d'experts". In Proceedings of InforSID 2012, May 29-31, Montpellier, France.

[9] C. Dutrey, A. Peradotto, and C. Clavel, "Analyse de forums de discussion pour la relation clients : du Text Mining au Web Content Mining", Actes JADT'2012, Liège (Belgique), 13-15 juin 2012.

[10] S. Evert, and E. Giesbrecht, "Part-of-speech tagging – a solved task? An evaluation of POS taggers for the Web as corpus.", Proceedings of the 5th Web as Corpus Workshop (WAC5), San Sebastian, Spain, 2009.

[11] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate Detection using Shallow Text Features", Third ACM International Conference on Web Search and Data Mining (WSDM 2010), New York City, USA.

[12] S. Lohmann, J. Ziegler, and L. Tetzlaff, "Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration", Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT '09), pp. 392-404.

[13] M. Marrero, S. Sánchez-Cuadrado, J. Morato, G. Andreadakis, "Evaluation of Named Entity Extraction Systems, Research", Computing Science, Vol. 41, 2009, pp. 47-58.

[14] C. Martineau, E. Tolone, and S. Voyatzi, "Les Entités Nommées : usage et degrés de précision et de désambiguïsation". In Catherine Camugli Gallardo, Matthieu Constant, and Anne Dister, editors, Actes du 26ème Colloque international sur le Lexique et la Grammaire (LGC'07), Bonifacio, France, Octobre 2007, pp. 105-112.

[15] F. Pirolli, "Apports des folksonomies dans le cadre d'un processus de veille : vers la prise en compte des spécificités informationnelles", ISKO-France, Lyon, France, 2009.

[16] B.A. Ribeiro-Neto, A.S. da Silva, J.S. Teixeira, "A brief survey of web data extraction tools", ACM SIGMOD Record, Volume 31 Issue 2, June 2002, pp.84-93.

[17] M. Roche, T. Heitz, O. Matte-Tailliez, and Y. Kodratoff, "EXIT: un système itératif pour l'extraction de la terminologie du domaine à partir de corpus spécialisés", JADT'04, Mars 2004, Belgique.

[18] S. Sekine, and C. Nobata, "Definition, dictionaries and tagger of Extended Named Entity hierarchy", Proceedings of LREC, 2004.

[19] K. Toutanova, D. Klein, C. Manning, and Y. Singer, "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network", Proceedings of HLT-NAACL 2003, pp. 252-259.

[20] M. Wilkens, "Evaluating POS Taggers: Speed", November 8, 2008, http://mattwilkens.com (read: January 31, 2014).

[21] M. Wilkens, "Evaluating POS Taggers: Coda", February 9, 2009, http://mattwilkens.com (read: January 31, 2014).

[22] Z. Zhang, J. Iria, C. Brewster, and F. Ciravegna, "A Comparative Evaluation of Term Recognition Algorithms". In Proceedings of The sixth international conference on Language Resources and Evaluation (LREC 2008), May 28-31, 2008, Marrakech, Morocco.

Towards the Design of a Textile Chemical Ontology

Carolina Prieto Ferrero
Chemical Department
AITEX, Textile Research Institute
Alcoy, Spain

Elena Lloret
Department of Software and Computing System University of Alicante Alicante, Spain

Manuel Palomar
Department of Software and Computing System University of Alicante Alicante, Spain

***Abstract*—The main goal of this paper is to present the initial version of a Textile Chemical Ontology, to be used by textile professionals with the purpose of conceptualising and representing the banned and harmful chemical substances that are forbidden in this domain. After analysing different methodologies and determining that "Methontology" is the most appropriate for the purposes, this methodology is explored and applied to the domain. In this manner, an initial set of concepts are defined, together with their hierarchy and the relationships between them. This paper shows the benefits of using the ontology through a real use case in the context of Information Retrieval. The potentiality of the proposed ontology in this preliminary evaluation encourages extending the ontology with a higher number of concepts and relationships, and validating it within other Natural Language Processing applications.**

***Keywords*—Ontology; Textile Chemical; Chemical Ontology; Textile Ontology; Textile Chemical Ontology; NLP**

I. Introduction

The information available on the Web is increasing at a fast rate. This can be viewed from a double perspective: it is positive, since users can be more and more informed, but in contrast, the negative aspect is that information can be overwhelming and users cannot manage it in an effective and efficient manner. In particular, when we deal with information of a specific domain, this problem is exacerbated.

Specifically, this occurs to the professionals of the Textile Chemical domain that constantly need to be up-to-date concerning all the directives and legislation about harmful substances applied to the Textile Chemical domain. As it was shown in [9] general purpose tools, such as Google or Yahoo search engines, are not sufficient for the type of information they need to deal with. For this reason, specialised tools and resources capable of facilitating the processing and understanding of this type of information would be crucial. In this sense, Natural Language Processing (NLP) can provide tools and resources in order to retrieve, extract, classify, summarise the information of interest in a specific domain. Moreover, when focusing in very specialised domains (e.g., medicine, chemistry, etc.), specific knowledge is also needed.

In order to be able to represent semantic knowledge, ontologies can be developed. In [1], the term ontology is described as the basic concepts and relations comprising the vocabulary of a topic area, as well as the rules for combining concepts and relations to define extensions to the vocabulary. An ontology can be also defined as "a formal and explicit specification of shared conceptualization" [12].

The difference between ontologies and taxomonies is that taxonomies are constituted by strict subclasses from the relationship "is-a" [22]. Moreover, ontologies are a powerful semantic knowledge representation that have been used in a wide range of contexts. In particular, the use of ontologies have been proven successfully in several domains like health domain [20], pharmacological [22], tourism [21], agriculture [23], e-business [24], education [25], or chemistry [9, 26] among others.

In particular, in the Textile Chemical domain we want to represent the knowledge about the banned and harmful substances. For this, it is necessary to develop an ontology containing these substances because textiles are subjected to strict controls according to the current directives that are constantly being updated. Additionally, to ensure the safety of the final consumer, when a laboratory detects any banned chemical substance, the product is removed from the market.

Therefore, the aim of this paper is to present the initial version of an ontology in the Textile Chemical domain. More specifically, the goal will be achieved by first gathering the expertise and knowledge of the professionals in the Textile Chemical domain, with the purpose of determining the relevant concepts and relations for developing and building the Textile Chemical Ontology. Then, with this knowledge, the ontology will be developed, further populated, and finally, it will be evaluated.

The validation of the initial version of the ontology will be carried out in the context of information retrieval, through a set of preliminary experiments with some relevant concepts and instances (without using the whole ontology). General-purpose search engines with and without using the ontology concepts and relations will be compared, determining whether the use of the ontology will retrieve more specific results.

This paper is organised as follows. In Section II, the related work is briefly described. Section III describes the design of the Textile Chemical Ontology. Section IV presents the use case where the ontology will be validated in a preliminary manner. Finally, Section V concludes the paper, and outlines the future work for developing the ontology completely.

II. Related Work

This section focuses on existing ontologies that could be related to some extent to the one proposed in this paper. The analysis of the different methodologies employed can be found in Section III.A.

To the best of our knowledge, at the present moment, it there is not any ontology for the Textile Chemical domain. Only the Project SEAMLESS[1] with TEX GLOB as a textile ontology [7] has been found. TEX GLOB ontology consists of three parts: the TEXTILE TAXONOMY, the TEXTILE VOCABULARY and TEXTILE DATA MODEL. The TEXTILE VOCABULARY, named as TEX GLOB vocabulary, is a tool where the relevant textile terms can be found with their explanation, and it indicates if a term belongs to the taxonomy or to the data model.

The terms and definitions were extracted from the CORE ontology and one part is used for the TEX GLOB data model [27]. As the CORE ontology, the TEX GLOB data model is split into some parts to represent, the vocabulary and taxonomy concepts, respectively, as well as the main data types generated by the companies, namely company profiles, products/services, and business documents. One part is unchanged and taken as it is from the CORE Ontology whereas other parts are defined by importing the homologous CORE parts and then extending them with the addition of attributes and classes.

Other interesting ontologies are the chemical ontologies where CO [5] or ChEBI[2] can be found, among others. On the one hand, CO is a chemical ontology for the identification of functional groups and semantic comparison of small molecules. This is a small ontology based on the assignment of functional groups through a computerised tool called Checkmol. One advantage to note is that the terms are assigned automatically by the program and connected with a chemical structure and a definition.

On the other hand, ChEBI contains a small set of molecules belonging to chemical compounds. The concepts included in ChEBI represent natural and synthetic products that could be found in the internal process of organisms, together with the types of entities.

Moreover, it includes an ontological classification with the relationships between molecular entities and their parents and/or children [4], as well as the relations with other ontologies [2].

Moreover, interesting databases related to the Textile Chemical domain are ChemTop[3] and Chem-BLAST[4]. With ChemTop, it is possible to query about chemical and physical properties of the chemical species. Many data are collected from NIST[5] Website. In this database all the information is structured from the chemical compounds perspective.

In contrast, ChemBLAST is an incomplete database of compounds where the user can find structures in 2-D. This could be of interest only for very specific Chemistry context.

As it has been shown before, there is not any Textile Chemical Ontology available (neither a specific database) for the professionals working in this domain, so they have to manually cope with all the information they need, without using specific NLP tools.

For this reason, a new Textile Chemical Ontology is proposed. Its aim is to model the knowledge of harmful and banned substances in the Textile Chemical domain. Not only will be the Textile Chemical Ontology novel, since it will focus on harmful substances and components that could be used in textiles, but also it will add value for the research community, allowing to have all the related knowledge represented in an ontology, thus being useful for employing or integrating it in NLP tasks.

III. Desing of the Textile Chemical Ontology

The aim of this section is to analyse and justify the methodology used for building the Textile Chemical Ontology, as well as to describe the concepts and the types of relationships our ontology covers in its initial version. Therefore, an analysis of the existing methodologies is first provided (Section III.*A*), then the methodology employed for the development of our Textile Chemical Ontology is explained (Section III.*B*), and finally to what extent the information existing ontologies described in Section II can be reused, adapted or extended is discussed (Section III.*C*).

A. Analysis of Existing Methodologies

In [3] different methodologies for building ontologies are described. Some of the most popular ones include:

- CyC [14]
- Uschold and King [18]
- Grünninger and Fox [13]
- Kactus [15]
- Methontology [19]
- Sensus [17]
- On-To-Knowledge [16]

The use of one methodology or another will depend on different factors concerning the development of the ontology. Among these factors one can find: the knowledge acquisition, the verification and validation of the process, or the documentation in the integral process.

In the process of the ontology development, other issues to consider are the requirements, design, implementation, and maintenance; all of them related to the part of the World to be represented.

In [3], an extensive analysis of the different methodologies for designing and developing ontologies is carried out. Based on this analysis and our findings, the methodologies of CyC, Uschold and Kind and Methontology have been chosen, because they are the most complete methodologies between all the analysed methodologies. In these methodologies the process of development and the use of the ontology are independents, and therefore, they could be the most appropriate for designing our Textile Chemical Ontology.

Next the stages involved in each of the methodologies previously chosen are described in more detail, in order to see the common points and differences between them.

The stages involved in ***CyC*** methodology are:

[1] http://www.seamless-ip.org
[2] http://www.ebi.ac.uk/chebi/
[3] http://webbook.nist.gov/chemistry/
[4] http://bioinfo.nist.gov/SemanticWebpr3d/chemblast.do
[5] http://www.nist.gov

- Extract the necessary knowledge from all the available and interesting information sources for our represented domain.
- Acquire new knowledge using NLP tools.
- Develop and represent the ontology.

The method proposed by ***Uschold and Kind*** consists of four phases:

- Identify the purpose of the ontology
- Build the model
- Evaluate the model
- Document the ontology

Finally, for building ontologies with ***Methontology*** methodology, it is necessary to:

- Specify the objectives and decide the domain
- Conceptualise the terms
- Formalise the model
- Implement the mode
- Maintain and incorporate more information

As it was described before, the stages of each methodology show the process of building an ontology according to each model.

The method proposed in Methontology is more complete than others, in the sense that this methodology have a life cycle of the ontology where the technicians can make proves until everything is finished and the ontology can be used. That is the main reason why this methodology was finally chosen for building the Textile Chemical Ontology.

B. Methodology Chosen: Methontology

From the analysis carried out in the previous section. Finally the "Methontology" was determined as the most suitable methodology to start developing the ontology.

This decision was also motivated by how Methontology is structured for building the Textile Chemical Ontology. The stages are very clear and better structured than others for building ontologies. Moreover, the flexibility of this methodology for building and ontology, allows us to adapt the Textile Chemical Ontology construction to our needs.

Since we are interested in representing knowledge about harmful or banned chemicals applied to textiles in any form, the stages that this methodology takes into account for building the Textile Chemical Ontology need to be followed. The Methontology methodology has been chosen by different authors (e.g., [8, 6,11]) for building the ontology, and we will follow the same guidelines for building ours. Next, the stages for building our ontology are described:

- ***Specification.*** This stage identifies the purpose of the ontology, domain of use and users, the degree of formality required, and the scope of the ontology including the terms that be represented.

In our particular case, the purpose of creating this ontology is to help researchers and professionals working in the Chemical Textile domain, when looking for information concerning new legislation that could affect textiles. Currently, they need supporting tools to find specific information very quickly and in short time, which unfortunately, they are not available yet.

- ***Knowledge acquisition.*** This stage is developed in parallel as the previous stage. Any type of knowledge source and any method can be used to build the ontology, although the roles of expert interviews and analysis of texts are very valuable. The knowledge required for developing the ontology will come from a professional of the Textile Chemical domain and the information sources describe in Section III.*C*.
- ***Conceptualisation:*** In this stage, the concepts, relations and properties are identified. Once the concepts, relations and properties are identified, they are represented using an applicable informal representation. After that, once the knowledge is conceptualised, the ontology can be populated with the corresponding instances. Further detail about this stage in Sections III.*C* and III.*D* is provided.
- ***Integration:*** In the event that more information is needed, the knowledge available in other ontologies can be reused to complete the information. As it was described in Section II, several and different types of ontologies related to either Textile or Chemistry can be found. They can be advantegous for this research in the sense that some information contained in existing ontologies could be reused or adapt. They can also be useful for extracting some concepts and making extensions for preparing the Textile Chemical Ontology. In this manner, we can take advantage of the provided taxonomy and vocabulary from the textile ontology (TEX GLOB).

Moreover, we can also reuse some information about the structures that are included in chemical ontology (ChEBI), since some of the concepts that we may need to include in our ontology could be already present in chemical ontology. The integration of other ontologies will increase the robustness of ours.

- ***Implementation.*** In this step, the ontology is represented in a formal language. In particular, for the proposed Textile Chemical Ontology, XML language will be used, because it is a standard that can be used to encoded the knowledge that could be then integrated in automatic processes. In addition, the ontology will be developed in English, because professionals working in the Textile Chemical domain normally use this language, although future extensions to other language would be also possible.
- ***Evaluation.*** After creating the ontology, it is necessary to evaluate the ontology by checking how complete or valid it is. At the moment, to verify the usefulness of the ontology, a use case in the context of the information retrieval is designed, which is explained in

Section IV. This use case will serve to analyse the usefulness of the proposed Textile Chemical Ontology in a preliminary way.

- ***Documentation.*** In order to be reusable and understandable by the research community or the group of professionals working with it, the ontology must be documented using a specific software for building ontologies, e.g. PROTÉGÉ[6]. These documents will contain all the information about the design and development of the ontology.

All the previously mentioned steps, except the information integration, have been already completed. Thus, this constitute the first cycle of the ontology. The ontology life cycle answers the previous questions identifying the *set of stages* though which the ontology moves during its life [19]. Making an analogy, it could be said that the ontology development process is similar to the production chains in a manufacturing domain as the ontology is to the final product that such production chain. Later, we could add more information to the ontology if newer concepts or instances are obtained, so that a better and more complete ontology is created. For building the Textile Chemical Ontology this life cycle in our methodology will be taken into account.

When all the steps are finished the life cycle of an ontology is closed and is prepared for using by the users.

C. Concepts and Properties Definition

At the moment, in the initial version of our Textile Chemical Ontology, it has been structured in concepts and relationships.

Concepts are the main basic piece of the knowledge to represent, and the instances are more specific concepts represented in ontology. Instances will be derived in the ontology population, which is out of the scope of this research work. Relationships represent the interaction between concepts and shape the structure of domain.

For extracting the concepts that will be part of Textile Chemical Ontology, the knowledge sources shown in Table I are used. These Websites contain all the legislation from Europe and other countries applied to the textile domain and therefore, banned chemical compounds also appear, such as *Lead*. In the case of the REACH Web, some chemical products from other domains as environmental, food, plastics, wood and others, can be also found, but only the chemical compounds applied to the Textile Chemical domain will be employed.

These information sources will be used to extract the necessary concepts for building our proposed ontology. However, to start with, a set of concepts were defined, and later grouped into different levels. The reason to choose these initial concepts was because all of them are already legislated and banned.

The proposed 3-level structure will allow us to define common properties to all levels (defined in the superclass), as well as specific properties only for the different subclasses.

[6] http://protege.stanford.edu/

For instance, **chemical substance** (first level), **heavy metals** and **chemical residues** (subclases of chemical substance; this is a second level) and **DMFu** (dimethylfumarate) and **SCCP** (Shot chain chlorinated paraffins) (subsubclases of chemical substance; this is a third level).

TABLE I. OVERVIEW OF WEBSITES FOR BANNED CHEMICAL COMPOUNDS

WEBSITES	*URL*	*EXAMPLE OF BANNED COMPOUNDS*
OEKO-TEX®	***http://www.oekotex.com***	***Lead***
REACH	***http://www.reachinnova.com***	***Lead***
AAFA	***https://www.wewear.org***	***Lead***

Figure 1 shows a fragment of the concepts of the Textile Chemical Ontology. In this fragment the concepts according to the three levels of hierarchy previously mentioned are represented.

- Chemical Substance
  - Heavy Metals Extractables
    - Arsenic
    - Lead
    - Cadmium
    - Chromium
    - Nickel
    - Mercury
  - Chlorinated Phenols
    - Pentachlorophenols
    - Tetrachlorophenols
  - Organic Tin Compounds
    - Tributyltin
    - Triphenyltin
    - Dibutyltin
    - Dioctyltin
  - Phthalates
    - Di-iso-nonylphthalate
    - Di-n-octylphthalate
    - Di-isodecylphthalate
    - Di-(2-ethylhexyl)-phthalate
    - Butylbenzylphthalate
    - Dibutylphthalate
    - Di-iso-butylphthalate
    - Di-C6-8-branched alkyphthalates
    - Di-C7-11-branched alkyphthalates
    - Di-n-hexylphthalate
    - Di-pentylphthalate
    - Di-(2-methoxyethyl)-phthalate
  - Chemical Residues
    - 1-Methyl-2-pyrrolidone
    - N, N-Dimethylacetamide
    - Dimethylformamide
    - Dimethylfumarate
    - Short Chain Chlorinated Paraffins
    - Tris (2-chloroethyl)phosphate

Fig. 1. Fragment of the concepts of the Chemical Textile Ontology

D. Relationships Definitions

After extracting the concepts from different knowledge sources, the next step in the design process is to establish the relationships between them.

Some relations that can be found in the domain are: -is a-, -part of-, -contained in-, -affects-, -related to-, -cause-. At the current version of the ontology design, 3 types of relationships were identified: i) is-a; ii) cause; and iii) part-of.

Next, each of this type of relationships is illustrate with an example:

*LEAD **is a** Heavy Metal.*
*Phthalates **cause** cancer.*
*Chemical Residues **part of** Chemical Substance*
In the short-term, we will extend the number of concepts and therefore, identify more relations between them.

IV. Use Case: Information Retrieval

In this section a use case where the Textile Chemical Ontology could add value to the final results is described. In particular, the scenario focus on the information retrieval task. Firstly, it is explainedhow the ontology could be used, and then a preliminary comparison with and without the use of our ontology is conducted.

In [10], it was shown that general-purpose alert systems as Google and Yahoo! Alerts were not suitable for searching highly specialised information. Several problems were encountered when using these systems, for instance, the problem of ambiguity. Concepts, such as *lead* or *flame* have different meanings, but Google and Yahoo! used to place at the first positions, the documents referring to their most frequent word senses, so very few results were provided about their meaning in the specific Textile Chemical domain. The results related to this domain, if retrieved, were always placed at the end of the list of retrieved documents, which was very difficult to find, given the high number of results that were retrieved.

In light of these experiments, a preliminary evaluation is performed, where the ontology is used to expand the terms of the query in order to analyse to what extent the retrieved documents could be more accurate. This term expansion could not be done without the knowledge about the Chemical Textile domain represented in our ontology.

For doing these experiments, 10 concepts about different levels from the Textile Chemical Ontology were used. The general-purpose Google search engine was selected for performing the searches, and each of these concepts was searched for individually (without exploiting the knowledge in our proposed ontology). Later, they were again searched but this time, using the relationships "is-a" of our ontology in order to expand the terms in the query, given an initial concept.

Table II shows the results of the experiments. We show the differences between using and not using the Textile Chemical Ontology. In both cases, a general-purpose search engine is employed.

After analysing the data obtained, it can be observed that the use of the Textile Chemical Ontology for performing the search helps to reduce the number of retrieved results. Moreover, analysing in detail whether the results among the 10 first positions may or not be related to the Textile Chemical domain, it was found that the ontology is also suitable to focus on more specialised documents.

Within this process, the 10 first documents that recover the general-purpose system were analysed in more detail. In the case of the *phthalates*, these documents are interesting for us without using the ontology, but when the Textile Chemical Ontology is used, the number of documents retrieved are lower but more specific in our domain. In the case of other terms, such as *Heavy Metals*, searched without the Textile Chemical Ontology, the 10 first documents are referred to the musical genre, but when the ontology is used, more specific and specialised documents, and therefore more interesting for our purposes are found.

When a search with general-purpose search engines is performed, the results obtained are general and many times they are not related to our domain. However, when the Textile Chemical Ontology is used for searching the information, the results obtained clearly belong to the specific domain. Therefore, the combination of NLP tasks, such as information retrieval and ontologies is very appropriate, since it is possible to retrieve specific information for the Textile Chemical domain, decreasing also the time spent for retrieving information of interest in this domain.

TABLE II. Validation Results (For Concepts) For The Ontology Using A General-Purpose Search Engine

CONCEPTS	*GENERAL SEARCH (Results)*	*SEARCH WITH THE ONTOLOGY (Results)*
Phthalates	***3,370,000***	***643,000***
Heavy Metals	***190,000,000***	***4,350,000***
Organic Tin Compounds	***6,730,000***	***68,500***
Chemical Residues	***4,700,000***	***1,710,000***
Chlorinated Phenols	***609,000***	***74,400***
Lead	***317,000,000***	***452,000***
Pentachlorophenol	***743,000***	***173,000***
Tributyltin	***378,000***	***76,700***
Cadmium	***28,100,000***	***5,600,000***

This use case was just an example of a scenario where the ontology could be validated. Although the experiments conducted are very preliminary, the comparison made reveals the potentiality of the ontology when apply to NLP tasks.

V. Conclusions and Future Work

In this paper, the initial design of an ontology for the Textile Chemical domain are presented and discussed. After the study of the related existing ontologies, no ontology specifically developed for the Textile Chemical domain was found. There are ontologies either for Chemistry or Textile domains independently, CheBI and TEX GLOB, respectively, but there is none that combines both fields into the same ontology.

In order to start designing the Textile Chemical Ontology, select a suitable methodology. In this case, it was decided that "Methontology" methodology will be used to develop the ontology. Having analysed other available methodologies, "Methontology" is the best methodology for building the Textile Chemical Ontology, since this methodology is very clear and it provides a better organisation for the stages involved in the ontology development process.

Following the stages of this methodology, two key issues in any ontology are: i) the definition of concepts and ii) relationships. For the former the sources of information where the concepts constituting the ontology can be extracted (e.g., OEKO-TEX or REACH) were first analysed. For the latter, the different types relations were also studied, being three of them (is-a, cause, and part-of), the most relevant ones.

Finally, a use case where the Textile Chemical Ontology was used in the context of Information Retrieval task was presented. In this manner, it was shown how the problem of ambiguity for some concepts could be solved, when employed general-purposes search engines (e.g., Google). In this manner, it was shown that when using the Textile Chemical Ontology the problem of ambiguity disappears, reducing the number of retrieved documents as well as obtaining higher precision.

In the short term, as a future work, the size of the ontology will be extended, increasing the number of concepts and relationships. One aspect planned to be added when developing the ontology would be to take into consideration the multilinguality of the concepts, adapting those parts of the ontology where different languages may be involved. Moreover, its validation in the context of information retrieval will be broaden and improved, integrating it and experimenting with specialised crawlers that might improve the results for specific domain.

In the medium and long term, the use of the ontology in the context of other NLP tasks, such as text summarisation or information extraction will be also analysed.

ACKNOWLEDGMENT

This research is partially funded by the European Commission under the Seventh (FP7 - 2007- 2013) Framework Programme for Research and Technological Development through the FIRST project (FP7-287607). This publication [communication] reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein. Moreover, it has been partially funded by the Spanish Government through the Spanish Government through the projects "Análisis de Tendencias Mediante Técnicas de Opinión Semántica" (TIN2012-38536-C03-03) and "Técnicas de Deconstrucción en las Tecnologías del Lenguaje Humano" (TIN2012-31224) and by the Generalitat Valenciana (project grant ACOMP/2013/067).

REFERENCES

[1] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. (1991). Enabling technology for knowledge sharing. AI Magazine, 12(3):36-56.

[2] Batchelor, C. (2008). An upper-level ontology for chemistry. In *Proceedings of the 2008 conference on Formal Ontology in Information Systems: Proceedings of the Fifth International Conference (FOIS 2008)*, pages 195–207, Amsterdam, The Netherlands, The Netherlands. IOS Press

[3] Corcho, O., Fernández-López, M., and Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies: where is their meeting point? *Data Knowl. Eng.*, 46(1):41–64.

[4] Degtyarenko, K., de Matos, P., Ennis, M., Hastings, J., Zbinden, M., Mcnaught, A., Alc´antara, R., Darsow, M., Guedj, M., and Ashburner, M. (2008). ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic acids research*, 36(Database issue):D344–D350.

[5] Feldman, H. J., Dumontier, M., Ling, S., Haider, N., and Hogue, C. W. (2005). Co: A chemical ontology for identification of functional groups and semantic comparison of small molecules. *FEBS Letters*, 579(21):4685–4691.

[6] Fernandez-Lopez, M., Gomez-Perez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium*, pages 33–40, Stanford, USA.

[7] Lima, C., Bonfatti, F., Sancho, S., and Yurchyshyna, A. (2006). Towards an ontology-enabled approach helping smes to access the single european electronic market. In Ghodous, P., Dieng-Kuntz, R., and Loureiro, G., editors, *ISPE CE*, volume 143 of *Frontiers in Artificial Intelligence and Applications*, pages 57–68. IOS Press.

[8] López, M. F., Gómez-Pérez, A., Sierra, J. P., and Sierra, A. P. (1999). Building a chemical ontology using Methontology and the ontology design environment. *IEEE Intelligent Systems*, 14(1):37–46.

[9] Prieto, C., Fernández, J., Lloret, E., and Palomar, M. (2012a). Specialized Information Retrieval in the Context of the Chemical Textile Domain. 3rd World Conference on Information Technology (WCIT 2012), 14-16 Nov, 2012, Barcelona, Spain.

[10] Prieto, C., Lloret, E., and Palomar, M. (2012b). Análisis de la Calidad de la Información Recuperada por Sistemas de Alertas en el dominio Químico Textil. In *Proceedings of II Spanish Conference on Information Retrieval*.

[11] Corcho, O., Fernández-López, M., Gómez-Pérez, A., López-Cima, A. (2005). Building legal Ontologies with Methontology and WebODE.

[12] F.Baader, D.Calvanese, D.Mcfuinness, D. Nardi and P. Patel-Schneider. (2003) The description Logic Handbook. Cambridge, U.K.: Cambridge Univ. Press, ch.2.

[13] Grünninger M., Fox M.S., (1995) Methodology for the desing an evaluation of ontologies, in Workshop on Basic Ontological Issues in Knowledge Sharing.

[14] Lenat D. B., Guha R. V., (1990) Building Large Knowledge-Based Systems: Representation and Inference in the CyC Project, Addison-Wesley, Boston.

[15] Schreiber Ath., Wielinga B., Jansweijer W., (1995) The KACTUS view on the 'O' word. Technical Report, ESPRIT Project 8145 KACTUS, University of Amsterdam, The Netherlands.

[16] Staab S., Schnurr H.P., Studer R., Sure Y., (2001) Knowledge processes and ontologies, IEEE Intelligent Systems 16 (1) 26–34.

[17] Swartout B., Ramesh B. Swartout K., Ramesh P., Knight K., Russ T., (1997) Toward Distributed Use of Large-Scale Ontologies, AAAI Symposium on Ontological Engineering, Stanford.

[18] Uschold M., King M., (1995) Towards a Methodology for Building Ontologies, in: IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal.

[19] Fernández M., Gómez-Pérez A., Juristo N. (1997) Methontology: From Ontological Art Towards Ontological Engineering. AAAI Technical Report SS-97-06.

[20] Rohrer E., Motz R., Díaz A. 13E (2010) Ontology-Based Process for Recommending Health WebSites. IFIP AICT 341, pp 205-214.

[21] Descamps-Vila L., Casas J., Conesa J., Pérez-Navarro A., Gutiérrez I. (2011) Hacia la mejora de la creación de rutas turísticas a partir de información semántica. V Jornadas de SIG Libre.

[22] Romà-Ferri, M.T. (2009) OntoFIS: Tecnología ontológica en el dominio farmacoterapeutico. Tesis Doctoral. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante.

[23] Ho-Young Kwon., Sabine Grunwald., Howard W. Beck., Yunchul Jung., Samira H Daroub., Timothy A. Lang., and Kelly T. Morgan. (2010) Ontology-based simulation of water flow in organic soils applied to Florida sugarcane, Agricultural Water Management, Elservier. 112-122.

[24] Missikoff, M. and Taglino, F. (2003). SymOntoX: a Web-ontology tool for ebusiness domains. In Proceedings of Fourth International Conference on Web Information Systems Engineering. 343-346.

[25] Jin Tan Yang., Min Jey Hwang., and Yuan Fong Chu.. (2005). A Study on Searching and Recommending SCORM CPs by Ontological Support. In Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies.

[26] Paula de Matos., Rafael Alcántara., Adriano Dekker., Marcus Ennis., Janna Hastings., Kenneth Haug., Inmaculada Spiteri., Steve Turner., and Christoph Steinbeck. (2010). "Chemical Entities of Biologial Interest: an update", Nucleic Acids Research. Oxford University Press, D249-D254.

[27] Project IST-FP6-026476 SEAMLESS. (2007) "Small Enterprises Accessing the electronic Market of the Enlarged Europe by a Smart Service Infraestructure". TEX Sector Ontology.

A Simple Strategy to Start Domain Ontology from Scratch

Ivo Wolff Gersberg / Nelson F. F. Ebecken
COPPE
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil

***Abstract*—Aiming the usage of Domain Ontology as an educational tool for neophyte students and focusing in a fast and easy way to start Domain Ontology from scratch, the semantics are set aside to identify contexts of concepts (terms) to build the ontology. Text Mining, Link Analysis and Graph Analysis create an abstract rough sketch of interactions between terms. This first rough sketch is presented to the expert providing insights into and inspires him to inform or communicate knowledge, through assertive sentences. Those assertive sentences subsidize the creation of the ontology. A web prototype tool to visualize the ontology and retrieve book contents is also presented.**

***Keywords*— domain ontology; contextual approach; ontology; NLP**

I. Introduction

Since 1990s Domain Ontology was seen as a way to formally model a system's structure. An ontology engineer seeks to represent specific knowledge, analyse the most relevant entities (more general and abstract entities that can be subdivided into categories such as objects, processes, and ideas) and organizes them into concepts and relationships. The skeleton of ontology consists of a hierarchy of generalized and specialized concepts [1].

Undeniably, different currents are found on the discovery of textual patterns for building ontology. On the one side are the experts of Natural Language Processing (NLP) and reasoning, claiming that a semantic approach is mandatory for dealing with ontologies. One of challenges during the transformation from data to knowledge is the use of semantic instead of traditional Text Mining techniques [2].

On the other hand, those that advocates the use of simple Text Mining statistical techniques. NLP could be used for an understanding of the analysis or synthesis of texts and not necessarily for an understanding of the texts [3]. Therefore, the Information Retrieval area and the NLP began to share algorithms and statistical methods with the help of lexical dictionaries to provide answers for relatively elaborate subjects. Many scientists still contend that these statistical methods are inadequate for contextual knowledge extraction; however, for certain purposes, they are reasonably efficient [4].

It is clear that the line is quite blurry thought between a genuine NLP tool and a statistical tool. How to build domain ontology in a fast and easy way without the use of sophisticated semantic tools and that it serves from aid to the study of neophytes of a certain subject is the main objective of this work.

One of the significant challenges facing an ontology engineer is that, in most instances, he does not possess the specific domain knowledge that is the subject. He does not know which concepts in a scientific text are important and how to start talking with an expert. By overcoming this shortcoming, the proposed methodology intends to provide a preliminary and abstract rough sketch of interactions between terms obtained from unstructured data (domain classical books), using automatic techniques of Text Mining, Link Analysis and Graph Analysis, based on contexts. This first rough sketch is presented to the expert providing insights into and inspires him to inform or communicate knowledge to the ontology engineer and construct Domain Ontology using two hands.

Once the ontology building was based in a middle-out strategy [5], in which concepts were generalized and specialized, and ontologies are always a work in progress, the main task is identify concepts (terms) that can relate to provide short assertive sentences (identified by the expert). The ontology engineer alone would not be able to identify and build such sentences, but on terms that seem obvious to expert´s eyes, the task becomes easier. How to get these concepts, i.e., how to extract terms from a non-structured data and how to presents those in a suggestive way to an expert, is the goal to build Domain Ontology from scratch.

II. Building Domain Ontology methodology

The proposed methodology, focused on practicality, utilizes two different tools: *PolyAnalyst Data Analysis* (PA)[1] for Information Retrieval/Link Analysis and *Gephi Graph Visualization*[2] for Graph Analysis.

A. Corpuses and extracting concepts

The first task determined that the Domain Ontology will be created about a mathematical subarea: Fractal. The resources for obtaining the relevant concepts were chosen by the expert and were composed of nine classical books on mathematical fractal, with an average of 340 pages and a total of 680,000 words (after pre-processing). It was considered using all chapter contents of the books and also

[1] Megaputer Intelligence, Inc.
[2] Gephi Consortium, CDDL and GNU General Public License

using only the words within the indexes of the same books (as concepts suggested by the authors - virtual specialists). Two corpuses were built: the FRACTAL Corpus (148 documents originated from the 148 chapters of the adopted books) and the Index Corpus (nine documents originated from the indexes of the adopted books). The Text Mining techniques were applied to these two corpuses separately.

The task of *Concept Acquisition* and *Selection* reveals the terms (nouns) considered essential to fractal knowledge. The strategy for this phase involved approaches without an expert's presence, starting from a set of terms (unigram and bigram) originated from PA tool. Those terms were measured by a significance value to represent how different a word is in all the texts, a measure 'above' as an average from the simple word frequencies, which is compatible with the classical measure Tf-IDF — term frequency–inverse document frequency, an excellent example of a statistical index that gives quantitative answer as to whether a term is really worth being extracted [6]. The sets were normalized, ranked and pruned by a high threshold.

B. Contexts to build assertive sentences

Contexts are abstract objects and difficult to be defined [7], normally examples are offered, but every communication needs contexts because without contexts there is no meaning [8]. Let´s consider an example: give the words *generator* and the word *tree* to a person without fractal knowledge, so this person cannot think towards a fractal context. Probably he will say, "Ok, a machine that converts one form of energy into another using trees", not ecological and perhaps not an assertive sentence. But a person thinking inside fractal knowledge, immediately and without effort will say "a generator that has a line segment, as an initiator, will construct a fractal tree".

Various notions will be found about what context is and how to treat it formally. Ramanathan Guha, over McCarthy works, tried to give a concept of context: Contexts are objects in the domain, i.e., we can make statements about contexts, as in [9]. A specific formula to treat contexts is used by McCarthy and Guha, based on sentences of the form: *ist(c,p),* where *ist* stands for *´is true in´* and is to be taken as assertions that the proposition *p* is true in the *context c*. Considering the idea of construct assertive sentences, but not operations using Reasoning and First Order Language as McCarthy and Guha, our question is *how to construct assertive sentences?* The unstructured data of the books creates our universe of discourse and this data in a structure of a graph is presented to the expert, aiming for immediate recognition of meanings in a short assertive sentence. These short sentences will point the concepts and relations, of the ontology (taxonomic or non-taxonomic).

The first approach to construct the contexts used the Link Analysis technique from the PA tool. Using the sets of terms, generated in the previous section, as single-level taxonomy, we applied them to the corpuses to obtain representative labels for each of its documents. In this way, the documents were deconstructed to a few isolated terms (concepts). Similar to reduced labels documents, favouring a computational gain, these words were submitted to analysis using the Link Analysis technique, looking for correlation patterns for which the connections among the vertices of the graphs are measured by tension values. The undirected graph generated was shown to the expert, who manually identified possible contexts, like cluster of words with something in common. In a second moment, an automatic method using Graph Analysis technique, Community Detection, was used to identify possible contexts and to compare the expert results. Community Detection is more often used as a tool for the analysis and understanding the structure of a network, for shedding light on patterns of connection that may not be easily visible in the raw network topology, as in [10]. Gephi tool offers the possibility of mixing two different techniques: Community Detection and Laplacian Dynamics. The first one requires partitioning a network into communities of densely connected nodes, with the nodes belonging to different communities that are only sparsely connected. The quality of the partitions resulting from these methods is often measured using the so called modularity of the partition [11]. The second one introduce the stability of a network partition as a measure of its quality in terms of the statistical properties of a dynamic process occurring on the graph, instead of the structural properties [12]. The tension values of the Link Analysis were used to weigh the network links. To be fair, with the number of communities manually marked by the expert, this number was controlled by the resolution factor parameter given by the Laplacian dynamics method. Once in possession of contexts, the expert identified and constructed assertive sentences about fractal knowledge. These assertive sentences help the ontology engineer to build the ontology hierarchisation.

III. Ontology Visualisation

The construction and use of generic ontologies provides an alternative method for searching for and visualising a desired portion of knowledge. Instead of search concepts and documents based only on keywords, it is possible to search by *context.* A Web search engine prototype was created, allowing visualisation of the *contexts* of the ontology and document (chapter) retrievals, providing fractal neophytes with an initial path for their studies. The implementation, through the Thinkmap [3] tool based on Graph Theory, allows for a visualisation of the created ontology as an oriented graph among the concepts (vertices) and the relationships (edges). The document retrieval is based on relationships, revealing the FRACTAL Corpus's most relevant chapters through the well-known algorithm Vector Space Model (VSM).

IV. Results

The task of *Concept Acquisition* and *Selection* was automatically applied over the two corpuses separately, limited by the high threshold, giving two term sets. The expert also manually chose concepts over an unranked set, generated from the contents of the books, without any kind of pruning, only to check the performance of the automatic

[3] *Thinkmap* Visualize Complex Information — Thinkmap, Inc.

extraction. Considering the expert's choices and only using the indexes of the books, we obtained 48% and 23% of terms in common for unigrams and bigrams, respectively, i.e., regular results. However, if we think that the indexes of the books were only words, totally unstructured, without any sentence and very short file size, probably we can use them alone in some occasions. Using only the contents of the books, it was obtained 100% and 32% of terms in common for unigrams and bigrams, respectively. It was decided to aggregate the contents books set and the terms that was in the indexes set and not in the contents of the books set, obtaining 100% and 47% of terms in common with the expert´s choices for unigrams and bigrams, respectively. Therefore, the strategy for joining those terms of the book indexes and the term set of the contents had better performance; in other words, this strategy improved the results compared with only using Text Mining of the contents or only the scenario that included the indexes of the books.

Once in possession of the unigram and bigram terms as the final set of concepts (590 terms), the task of *context detection* was performed by Link Analysis and Detection Community.

A. Building contexts and the Fractal Domain Ontology

Over the final set of concepts, the Link Analysis technique generated an undirected graph. This graph was presented to the expert and he outlined, by hand, possible contexts of the big fractal context (blue colour in **Error! Reference source not found.**). Those contexts gave the idea of the most generic concepts of the Fractal Ontology and the concepts inside the contexts were used to construct the assertive sentences. Aiming an automatic process to identify the contexts, the Community Detection of Networks was applied over the Link Analysis results (Fig. 2). Numerous similarities were observed between the contexts manually signed by the expert in the graph of the Link Analysis and the contexts as communities in the network of the Graph Analysis (0). The contexts (communities) automatically detected was presented again to the expert. The collaborative job between the expert and the ontology engineer to construct the ontology is shown in TABLE III. , with few examples per context, but of course numerous other assertive sentences were created.

B. Comparison with a semantic space model

The results were also compared with a semantic technique. The BEAGLE Model was applied using a word similarity visualization tool, Word2Word[4] , where words are represented by high-dimensional holographic vectors. An environmental vector is created to represent the physical characteristics of words in the environment (e.g., orthography, phonology, etc.), whereas the memory vector represent internal memory for co-occurrence and position relative to other words using convolution and superposition mechanisms [13]. In this case, we used all the words of our universe of discourse (≈ 680,000 words), showing concepts in a distribution based in a similarity distance.

The intention was observe if one concept inside a context has his neighbours closer together in a semantic similarity distribution, i.e., closer together with higher similarity metric.

Based in the semantic space created, TABLE I. shows the top neighbours to fractal, power law, probability density and iteration (concepts extracted from different contexts found in Fig. 2). In this space we have all kind of words like nouns, verbs, *stoplist* words, etc., but if we check only for nouns, we can observe that some of them (highlighted) are also in his respectively context found. The highlighted common terms found are, indeed, the most important concepts pointed by the expert viewpoint to construct the assertive sentences.

TABLE I. TOP NEIGHBORS IN THE SEMANTIC SPACE

fractal		power law		probability density		iteration	
distance	to	distance	to	distance	to	distance	to
0.619	*selfsimilar*	0.378	fractal	0.360	*autocor-relation*	0.566	fractals
0.601	*fractal-dimension*	0.326	*scaling*	0.309	graph	0.561	*equation*
0.578	set	0.324	fractaldi-mension	0.294	nonde-creasing	0.551	*juliaset*
0.574	equation	0.322	probabi-lity	0.293	weier-strass	0.545	fractaldi-mension
0.567	probability	0.309	important	0.282	*brownian*	0.533	*set*
0.561	fixedpoint	0.302	integer	0.279	this	0.526	*point*
0.561	point	0.302	point	0.267	reason	0.526	*fixedpoint*
0.552	chaotic	0.300	simple	0.263	fractal	0.525	*cantorset*
0.545	important	0.293	propor-tional	0.262	convex	0.524	example
0.549	*measure*	0.293	set	0.257	cases	0.523	result
0.542	*fractals*	0.292	selfsimilar	0.256	result	0.522	follows
0.539	general	0.290	equation	0.253	note	0.520	cases
0.538	cases	0.289	equivalent	0.252	general	0.517	plate
0.535	follows	0.288	obtained	0.250	fractal-dimension	0.517	system
0.535	simple	0.288	problem	0.248	juliaset	0.510	sierpinski-gasket
0.532	example	0.288	follows	0.247	*probabi-lity*	0.507	dimensions
0.531	result	0.284	chaotic	0.245	suppose	0.504	*attractor*
0.527	juliaset	0.283	constant	0.245	similarly	0.502	however
0.524	*selfsimi-larity*	0.282	similar	0.245	write	0.494	on

Another way to see the semantic space is laying out the nodes using Multidimensional Scaling (MDS) algorithm according to similarity relationships (Fig. 5). For example, observing the 100 near similarity terms in the semantic space for concept *iteration*, it was found some important concepts (red nodes) that were also found in the *context approach*.

Others nouns (concepts) that are around the *iteration* concept were found in different contexts, but this fact is not a big problem because the goal is to construct Domain Ontology in a middle-out strategy. This suggest us (for future works) a way to link the contexts of our approach, i.e., construct assertive sentences using a concept from one context and another from other context.

[4] *Word2Word* (W2W) — Kievit-Kylar,B., Cognitive Computing Lab, Indiana University, USA

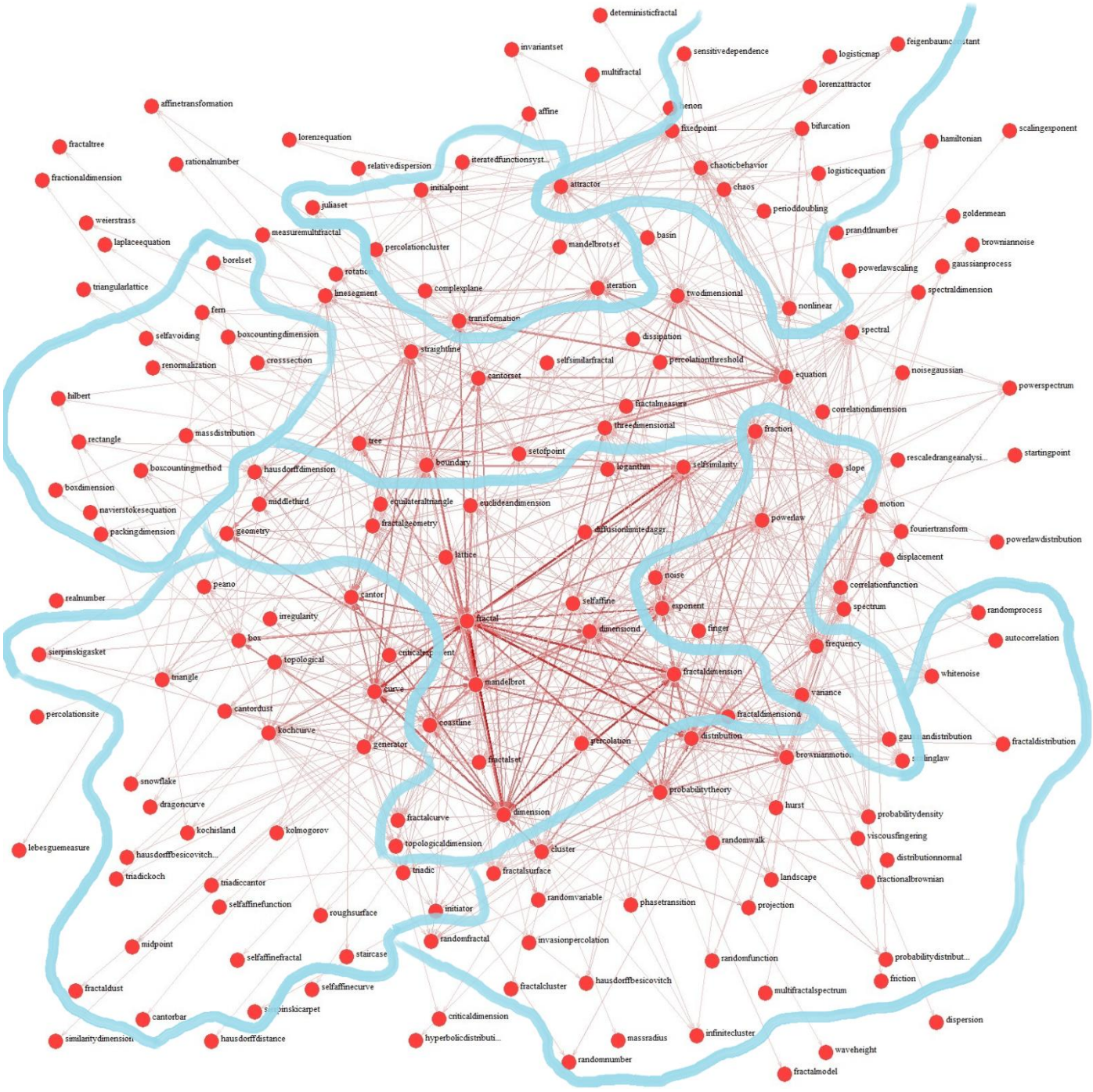


Fig. 1. Contexts Signaled by Expert over Link Analysis of Selected Term

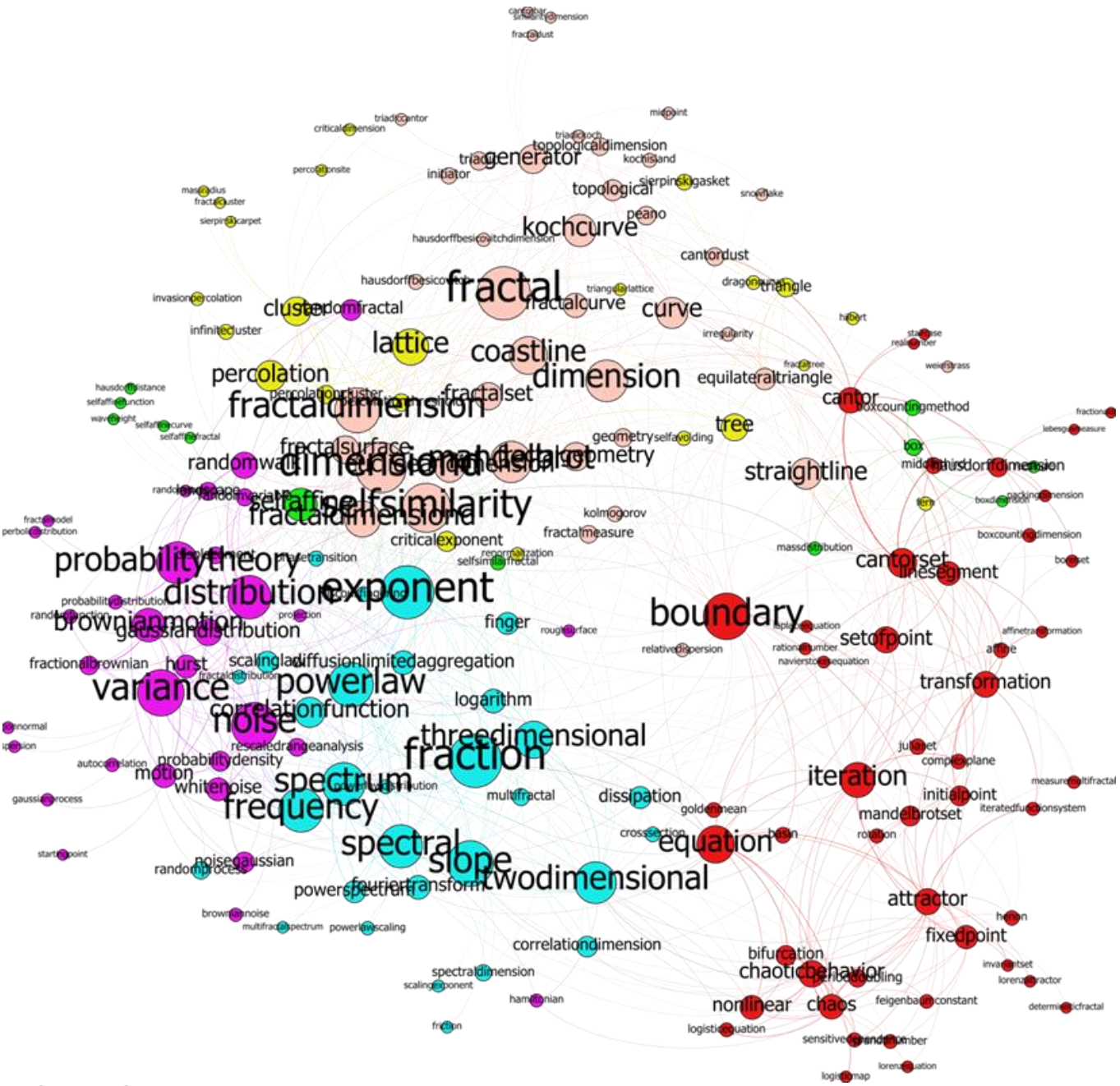


Fig. 2. Contexts as Community Detection

TABLE II. MANUAL X AUTOMATIC IDENTIFICATION OF *CONTEXTS*

manually *contexts* concepts		automatic *contexts* concepts	
upper right corner	attractor, nonlinear, chaos, chaoticbehavior , bifurcation, lorenzattractor, deterministicfractal sensitivedependence,	red *context*	attractor, nonlinear, chaos, chaoticbehavior , bifurcation, lorenzattractor, deterministicfractal, sensitivedependence,
upper middle	iteratedfunctionsystem, iteration, complexplane, mandelbrotset, juliaset	red *context* (middle)	iteratedfunctionsystem, iteration, complexplane, mandelbrotset, juliaset
upper left corner	borelset, boxcountingdimension, packingdimension, boxdimension	red *context* (upper)	borelset, boxcountingdimension, packingdimension, boxdimension
lower left corner	generator, initiator, curve, cantorbar, cantordust, snowflake, triadickoch, kochisland, triadic, midpoint, peano sierpinskigasket, dragoncurve	pink *context* (upper) yellow *context*	generator, initiator, curve, cantorbar, cantordust, snowflake, triadickoch, kochisland, triadic, midpoint, peano sierpinskigasket, dragoncurve
lower right corner	brownianmotion, fractionalbrownian, randomwalk, randomfractal, probabilitytheory, distributionnormal, whitenoise	magenta *context*	brownianmotion, fractionalbrownian, randomwalk, randomfractal, probabilitytheory, distributionnormal, whitenoise
right middle	powerlaw, exponent, scalinglaw, distribution, frequency	cyan *context*	powerlaw, exponent, scalinglaw, distribution, frequency, logarithm
middle	fractal, fractalgeometry, fractalset, mandelbrot, dimension, dimensiond, selfsimilarity, coastline, topologicaldimension, hausdorffdimension	pink *context*	fractal, fractalgeometry, fractalset, mandelbrot, dimension, dimensiond, selfsimilarity, coastline, topologicaldimension, hausdorffbesicovitchdimension

TABLE III. FEW EXAMPLES OF ASSERTIVE SENTENCES CREATED BY THE EXPERT

context	assertive sentence	ontology concepts	relation
pink	All generators are obtained by an initiator.	Generator –Initiator	hasInitiator
	Self-Similarity is one of the most important property of fractal objects.	Fractal – SelfSimilarity	hasProperty
magenta	Fractional Brownian motion is a random walk process.	FractionalBrownianMotion – BrownianMotion	is_a
		BrownianMotion – RandomWalk	is_a
cyan	Fractal Dimension is calculated by a Power Law function.	DimensionD – PowerLaw	isCalculatedBy
		PowerLaw – Function	is_a
	Power Law has a mathematical property that obeys a Scaling Law, scale invariance.	ScalingLaw – MathematicalProperty	is_a
		ScaleInvariance – ScalingLaw	is_a
		PowerLaw – ScaleInvariance	hasProperty
red	The Mandelbrot Set and the Julia Set are set of points in the complex plane, created by iteration process, Iterated Function System (IFS).	MandelBrotSet – SetOfPoint	is_a
		JuliaSet – ComplexPlane	isLocated
		Iteration – Process	is_a
		IFS – Iteration	is_a
	Rotation is a Linear Transformation.	MandelbrotSet – IFS	isCreatedBy
		JuliaSet – IFS	isCreatedBy
green	A Fractal can be a self-affine fractal or self-similar fractal; it means it can have an affine or a similar symmetry.	SelfAffineFractal – SelfAffine	hasSymmetry
		SelfSimilarFractal - SelfSimilar	hasSymmetry
		SelfAffine – AffineTransformation	hasTransformation
		SelfSimilar –LinearTransformation	hasTransformation

C. Ontology Visualisation Web Prototype

Once in the possession of the ontology, a prototype of searches oriented by contexts was implemented as a Web app, Fig. 3. The FRACTAL Domain Ontology visualization was created as an oriented graph among the essential concepts and the relations that clarify the fractal knowledge. The relationships of the taxonomic type are represented by grey edges, whereas the non-taxonomic relationships, which are knowledge in itself, are represented by pink colour. When passing the mouse on the relations, the tool indicates the specific name of the relation.

Fig. 3 enhanced the non-taxonomic relation *isCalculatedBy*, indicating that a fractal can be calculated by a power law. Clicking on a concept or looking for a certain concept (in the search area), leads to the tool unfolding new concepts related to the selected or searched for concept, presenting a context of relations appropriate to a certain desired granularity. The interface is easy to use, does not demand any previous knowledge, and is handled by clicking on concepts or on relations or by dragging on concepts. A sole concept does not clarify knowledge; thus, for the tool developed, only clicking on relations restores the chapters of the books.

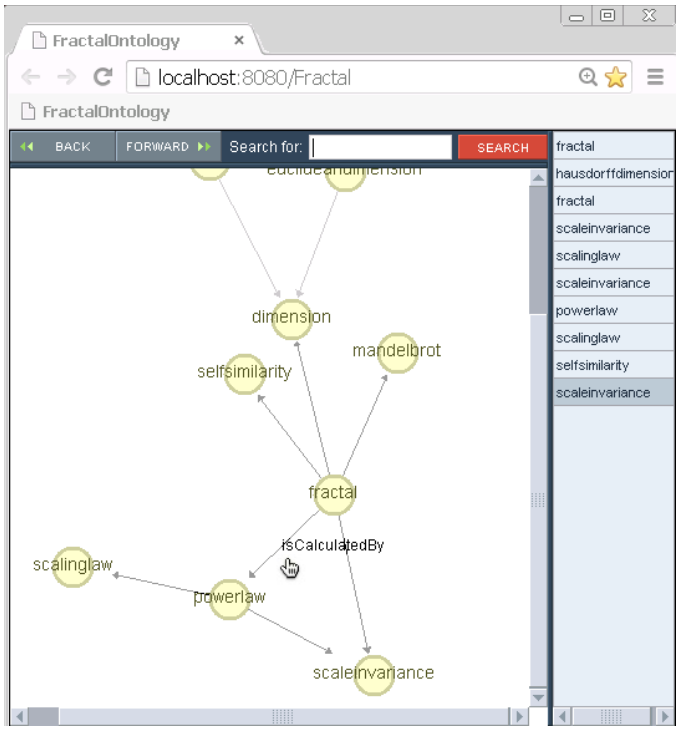


Fig. 3. A piece of Fractal Ontology Web Visualisation

When clicking on the enhanced relation, the tool presents an indication of bibliographical references for studies. The tool only returns the authors and the chapters of the books most relevant for associating the investigated relation, whereas the percentage relevance of each chapter is marked beside the selected chapter. The relevance is calculated as a modification of the Vector Space Model (VSM) technique, in which the vectors of the documents with a very small distance are considered similar to the relation. The more similar it is, the more relevant it will be in the document.

We now emphasise the notion of a contextual search in the following way: a student was interested in knowing where to learn how to calculate fractal dimensions through power laws. Using the example above (Fig. 3), a neophyte's glance into fractal knowledge by solely looking at the graph and moving the mouse onto the concepts, will see that fractals are calculated using a power law. Clicking on the relation *isCalculatedBy*, the chapters 1, 4, 5 and 17 of the book *Fractal, Chaos and Power Laws* was obtained as a result, whose author is Schroeder. The tool restricts the number of documents returned to those with a relevance of 99.5% or higher. For a student who is not curious and not stimulated by contextual reasoning, the task is concluded and he/she only studies those book or chapters. However, by consulting an expert, we found that among our nine books that have the most suitable chapters for understanding the calculus of fractal dimensions by power laws, he advised the following: chapter 2 of the book *Fractals* (author Feder) and chapter 4 of the book *Fractal, Chaos and Power Laws* (author Schroeder).

The usage of ontologies based on a search oriented by contexts will amplify the understanding of the subject. A student should be motivated to analyse the context in which the concepts *fractal* and *power law* are involved, as shown in Fig. 3. It is showed that the concept *power law* was verified to have a relationship with the concept *scale invariance* and with the concept *scaling law*; yet, the concept *fractal* was also related to the concept *scale invariance*. Within this context, we anticipate that the ontology relationship among the concepts *power law* and *scale invariance* is *hasProperty*. Therefore, if the student notices that the relation *hasProperty* is more intrinsic than *isCalculatedBy*, he/she will be urged to relationally analyse the concepts *power law* and *scale invariance*. Selecting this last relation, he/she will obtain Fig. 4 as a result.

Thus, the student notices that the most relevant chapters of this relation are those with a relevance of higher than 99.9%; among these are chapter 2 of Feder's book and chapter 4 of Schroeder's book, which the expert verbally recommended. In the event that the student chose these books or chapters for study, such choices would have agreed with the expert's indications without having consulted him. Once there are no problems with copyrights for certain books, clicking the desired chapter/book enables it to be consulted digitally.

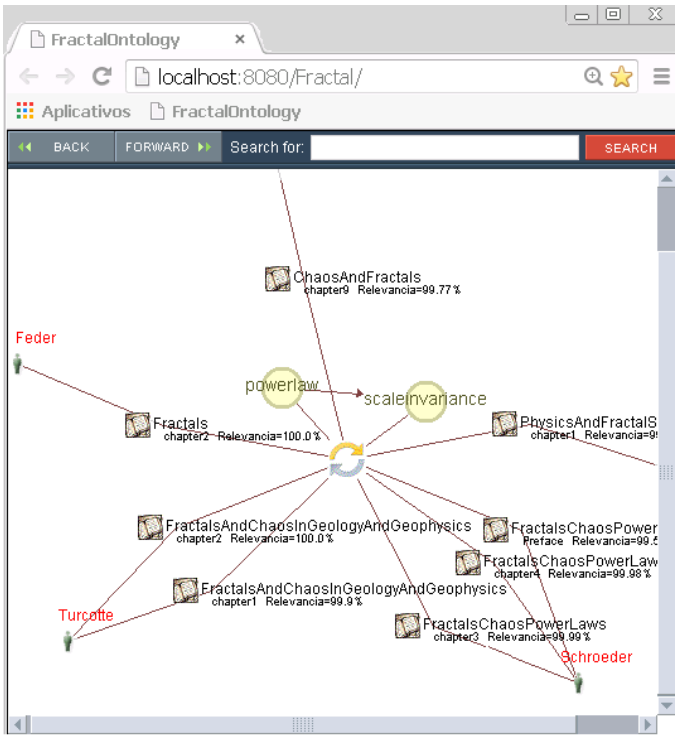


Fig. 4. powerlaw-scaleinvariance relationship (relevant chapters)

V. Final Considerations

The usage of ontology for us is not dedicated to reasoning, Q&A or merging databases. The main purpose, adopted in this work, was to build an educational Domain Ontology from scratch of the chosen subject in a fast and easy way. Based in classical books and having a vast consensual bibliography about the subject in question, the neophytes (students) can have an efficient retrieval method through a visual ontology web tool. In order to share knowledge, textual evidence needs to be linked to ontologies as the main repositories of represented knowledge [14].

A simple statistical approach looks for terms in a statistical way inside a text, while semantic information looks for terms whose grammar and syntax rules reveal some semantics. A semantic information need to know at least the phrase that contains the term in question, once words are characterized by the company it keeps [6]. Depending on the windows size of words to look at, the semantic approach will be time consuming.

The methodology used in this work gave attention to identify important terms together giving contextual meaning, using only simple statistical techniques (Tf-IDF and correlations of terms as a Link Analysis graph) and a network (communities) representation.

Emphasizing that the present methodology is a semi-automatic approach and accordingly to the fractal expert, the *contexts* created revealed us well an enough concepts to start a new ontology. A semantic distribution, like BEAGLE model, not offered great advantages in the present case.

Therefore, a simple way using a classical term extraction, Link Analysis and Community Detection can be used to start Domain Ontology from scratch, using only classical books about the subject in question.

ACKNOWLEDGMENT

We would like to give special thanks to Emeritus Professor Luiz Bevilacqua (COPPE/Federal University of Rio de Janeiro), who helped us through his notable knowledge of fractals. The authors acknowledge the support provided by CNPq, the Brazilian Research Agency, and FAPERJ, the Rio de Janeiro Research Foundation.

REFERENCES

[1] N. Guarino, D. Oberle, and S. Staab, "What Is an Ontology?," in *Handbook on Ontologies*, Springer-Verlag, 2009.

[2] H. Jingshan, D. Dou, J. Dang, J. H. Pardue, X. Qin, J. Huan, W. T. Gerthoffer, and M. Tan, "Knowledge acquisition, semantic text mining, and security risks in health and biomedical informatics," *World J. Bilogical Chem.*, vol. 3, no. 2, pp. 27–33, Feb. 2012.

[3] M. Konchady, *Text Mining Application Programming*. Massachusetts: Charles River Media, Thomson Learning Inc., 2006.

[4] Z. Li, M. C. Yang, and K. Ramani, "A methodology for engineering ontology acquisition and validation," *Artif. Intell. Fo Reng. Des. Anal. Manuf.*, no. 23, pp. 37–51, 2009.

[5] M. Uschold and M. King, "Towards a Methodology for Building Ontologies," presented at the IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, 1995.

[6] A. Stavrianou, P. Andritsos, and N. Nicoloyannis, "Overview and Semantic Issues of Text Mining," *SIGMOD Rec*, vol. 36, no. 3, pp. 23–34, Sep. 2007.

[7] J. McCarthy, "notes on formalizing context," 13-Apr-2005.

[8] G. Bateson, *Mind and Nature: A Necessary Unity*. New York: Dutton, 1979.

[9] R. V. Guha, "Contexts: a formalization and some applications," Stanford University, 1995.

[10] M. Newman, *Networks: An Introduction*, 1st ed. Oxford University Press, USA, 2010.

[11] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *arXiv:0803.0476*, Mar. 2008.

[12] R. Lambiotte, J.-C. Delvenne, and M. Barahona, "Laplacian Dynamics and Multiscale Modular Structure in Networks," *arXiv:0812.1770*, Dec. 2008.

[13] M. N. Jones and D. J. K. Mewhort, "Representing word meaning and order information in a composite holographic lexicon.," *Psychol. Rev.*, vol. 114, no. 1, pp. 1–37, 2007.

[14] I. Spasic, S. Ananiadou, J. McNaught, and A. Kumar, "Text mining and ontologies in biomedicine: Making sense of raw text," *Brief. Bioinform.*, vol. 6, no. 3, pp. 239–251, Sep. 2005.

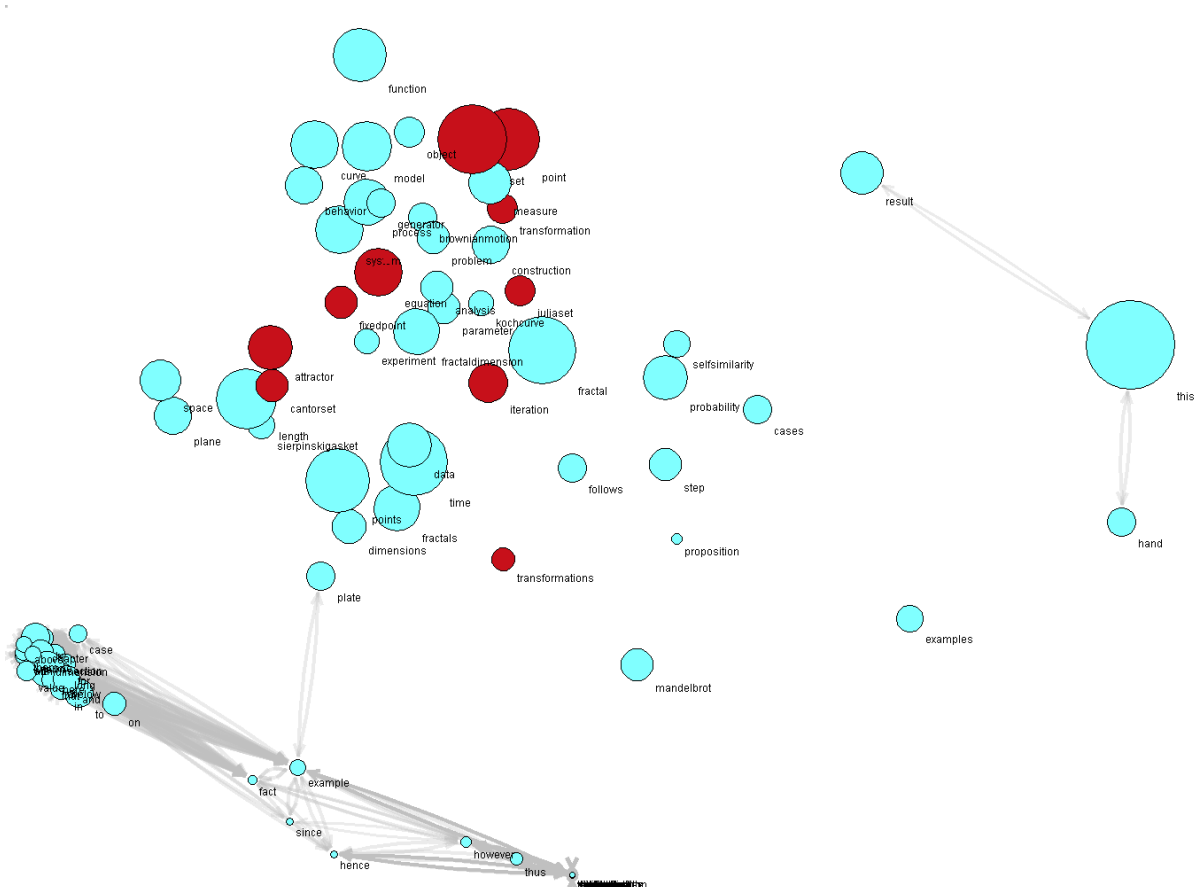


Fig. 5. Semantic Distribution of Concept *iteration*

An Effective Reasoning Algorithm for Question Answering System

Poonam Tanwar[1],
[1]Asst. Professor, Dept of CSE, Lingaya's University, Faridabad, Haryana, India & PhD Scholar, Uttarakhand Technical University, Dehradun,India

Dr. T. V. Prasad[2],
[2] Former Dean of Computing Sciences, Visvodaya Technical Academy, Kavali, AP, India.

Dr. Kamlesh Datta[3]
[3]Assoc. Prof & HOD (CSE), National Institute of Technology, Hamirpur, Himachal Pradesh, India

***Abstract*—Knowledge representation (KR) is the most desirable area of research to make the system intelligent. Today is the era of knowledge that requires articulations, semantic, syntax etc. These requirements, forced to design a general system which is applicable to represent declarative as well as procedural knowledge. Without effective inference/reasoning mechanism, the strength and utility of knowledge representation technique fulfill the partial requirement for an intelligent system. The objective of this research work is to present the effective/ appropriate knowledge representation technique for representing the general knowledge and a reasoning algorithm for Question Answering system (QAS) work as story reader, so that appropriate knowledge can be infer from the system. The architecture of knowledge representation system is capable to integrate different type of knowledge and it is cost effective also.**

***Keywords*—Knowledge Representation (KR); Semantic Net; Script; Reasoning; QAS; NLP**

I. INTRODUCTION

AI is the branch of science to make the machine as intelligent as human being for particular domain. Alternatively, it is the study of making machine intelligent by implementing intelligent programs to perform the complicated task. In 1950s, Alan Turing presented a paper on Computing Machinery and Intelligence. The result of this paper was if a machine could pass certain test (known as Turing test) then it could be intelligent. In this paper Turing also considered a number of arguments for, and objections to, the idea that computers could exhibit intelligence [1].

Knowledge representation (KR) is an essential area for cognitive science and Artificial Intelligence. In former, it is concerned with how knowledge is stored and processed, while in the latter the main aim is to solve problems requiring intelligence which otherwise is possible only through knowledge.

Broadly, KR is a study of methods of how knowledge is actually visualized/ realized and how efficiently/naturally it is similar to the depiction of knowledge in human brain. Constructing intelligent systems require large amount of knowledge and a method for representing large amounts of knowledge that permits interaction [1] [3]. KR is the fundamental issue in AI that attempt to understand intelligence [1] [2].

The main problem of AI system is how to represent knowledge and how to incorporate both types of knowledge in single system i.e., declarative and procedural [1]. Due to these issues, KR became a separate research area in AI. Since last few years a group of (two or more) methods are being considered as hybrid KR system that can address all these fundamental issues. KR techniques can be used for representing the knowledge required for Question Answering system.

a) KR Techniques

The KR techniques are divided in many categories. The representation techniques can be declarative, procedural, hierarchical, graphical, etc. Objects, properties, [17] categories and relations between objects, situations, events, states and time [17], causes and effects are the things that an intelligent system desires to represents [4][8]. The semantic net, conceptual dependency and script KR technique are described here. Semantic Net is commonly used KR technique that represents the connection between objects or class of objects. It is a directed graph in which nodes / vertices represent the objects/ class of objects and edges and links (unidirectional) represent the semantic relations between the objects. Semantic net are used to represent the inheritable knowledge. Inheritance is most useful form of inference. Inheritance is the property in which element of some class inherit the attribute and values from some other class [5][6][9].

The variant of semantic net i.e., partitioned semantic net can be used to delimit the scopes of quantified variables. [6][9].CD was developed by Roger Schank in 1973 to represent the knowledge acquired from natural language input. In CD Sentences are represented as a series of diagrams depicting actions using the abstract and real physical situations. CD representation provides the sets of primitive actions, different types of states, and different theories of inference. A variation in the theme of structured objects called scripts was devised by Roger Schank and his associates in 1973[5] [9][10]. Frame KR technique is also widely used based on object oriented concept. Many hybrid KR techniques also came for getting the advantage of KR techniques KL-ONE KR tool was the first hybrid KR technique which is the hybrid of semantic net and first order predicate logic. The FRORL, RT-FRORL are hybrid KR techniques [4][12][17[20]. In SOL, Hybrid KR the concept of smart object was used with encapsulation.

b) Question Answering System

An artificial QAS can be made for various applications like search engine, Natural language processing, Machine learning. This section presents a survey of various QAS.

i) **CNLP AIDE**: It is a off line system, based on four modules, question-answering system, document processing, Language-to-Logic (L2L), Search Engine, and Answer Providing Passages. Document processing is done offline, shown in figure 1. When a question is submitted by the user, it directly sent to the Language-to-Logic module, that generates the L2L query representation. The Search Engine module then searches the index and returns the top 200 relevant passages [21].

ii) **TREC-9:** It was the extension of TREC-8 and was based on a combination of the Okapi retrieval engine, Microsoft's natural language processing system and a module for matching logical forms. The questions was analyzed by NLPWin to produce a logical form. The query was contain the words like what, who, how, etc and stop words. Okapi IR engine was used for query term; the BM25 was used to retrieve weighting list of documents. The documents were segmented into sentences architecture of the system, Figure 2 [22].

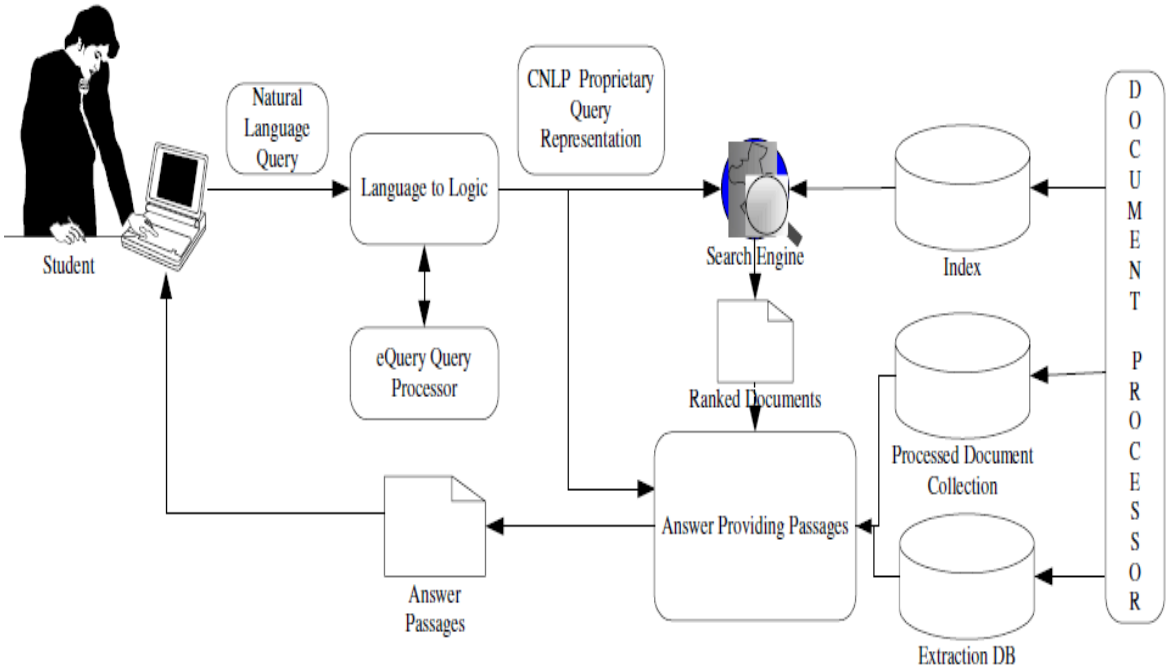


Fig. 1. Architecture of CNLP AIDE [21]

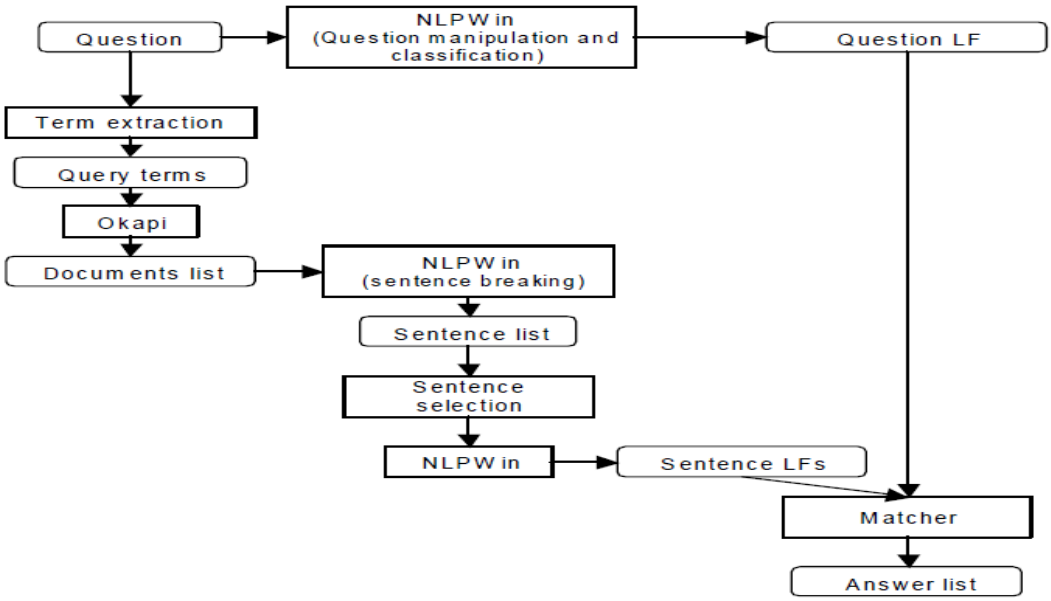


Fig. 2. Architecture of the TREC-9 [22]

c) *IQAS: It was proposed in year 2011 with architecture as in Figure 3. IQAS was designed to help the students so that they could become the good reader, it process the input in NLP and provide the result with proper feedback. The performance of IQAS is based on the no of documents, the information that user need and relevance judgments [23].*

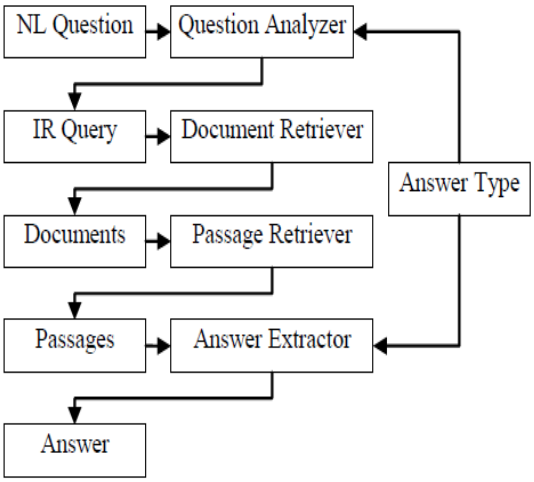


Fig. 3. represents the architecture of IQAS [23]

II. Knowledge Base System Architecture

The research in AI is divided in to two categories KR and general. For making the computer or machine as intelligent as human being, it requires two things KR and inference mechanism. Development of an AI system is a crucial task because some time we have incomplete information and it can be ambiguous and uncertain. So solution to these problems is to build a knowledge effective knowledge base and an effective inference mechanism.

a) Knowledge Base (KB)

The knowledge base used in Figure 4 and Figure 7 is used to store the knowledge required to solve the problem domain. The KB in Figure 4 is used to store the incoming knowledge i.e., story and the hybrid representation corresponding to that story whereas the KB used in Figure 6 is used to store the rules required to inference the knowledge from the input.

The KR system must be able to represent any type of knowledge, "a) syntactic, b) semantic, c) logical, d) presupposition, e) understanding ill formed input, f) ellipsis, g) case constraints, and h) vagueness". For making it more effective the knowledge representation model is divided in to five sub parts the K Box, knowledge base, query applier, reasoning and user interface as shown in Figure 4.[3][8].

The knowledge base architecture defined in Figure 4 is used as a story reader. The knowledge base of the system is capable to store the knowledge which is a hybrid of semantic net and script. Semantic net is used to represent the inheritable and relational knowledge whereas semantic net is used to represent the events in the story shown in Figure 5. The methodology used to implement the system is given in Figure 6.

The source of input can to the system can be a book, newspaper, magazine etc. a check is made whenever a new input is entered by the user to see whether the same is already stored in knowledge base or not. If the same is already stored in knowledge base then system displays the alert message or else it accepts the new input and passes on for further processing. As shown in Figure 5 the system is able to take the input from outside word.

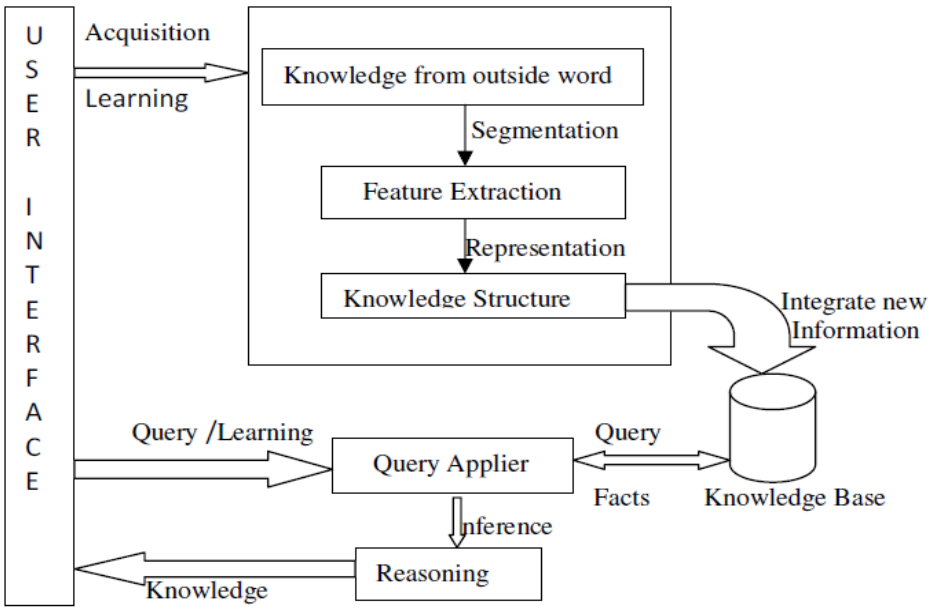


Fig. 4. Knowledge Base System Model / Architecture [3][8].

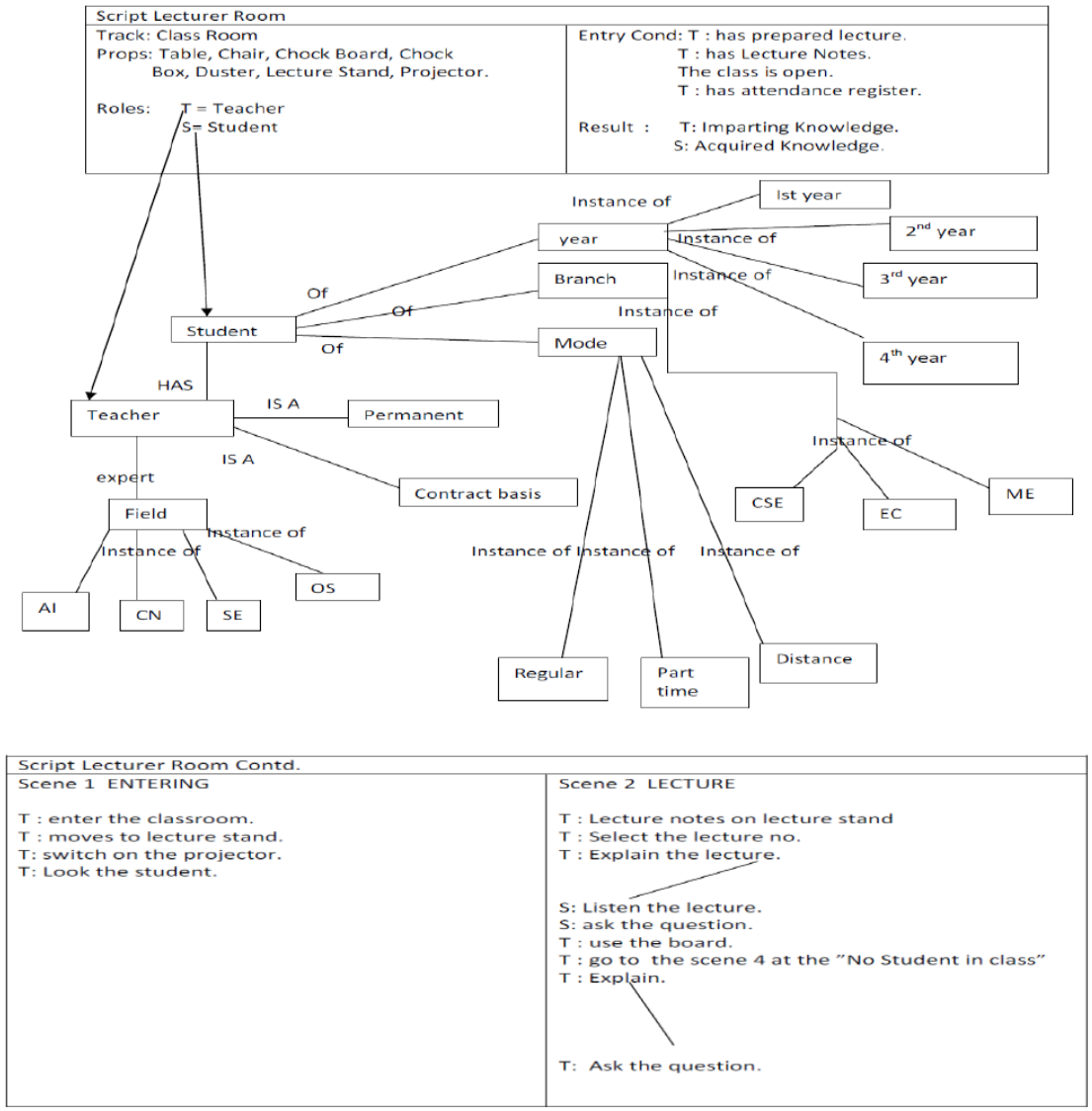


Fig. 5. Logical view of Hybrid KR [3][8].

b) Hybrid KR

The Hybrid KR used was the hybrid of semantic net and script KR techniques. The logical structure is shown in Figure 5. The system generates the semantic net for each input and links the semantic net with script of the story. The Hybrid KR was used to visualize the knowledge required for class room as well as the knowledge about students. For example, if any one wants to know the details of students whether he/she is 2nd year student and currently studying in semester 4 then the system can represent this aspect using semantic net and to visualise the interconnections that took place in the context of the class room, semantic net must be used. After generating the hybrid structure, a better detail will be available as to when the student will finish his/her degree. This hybrid structure proved to be very efficient in any situation where inheritable as well as stereotype knowledge was required.

c) Query Applier

Query Applier is used for obtaining the facts from the system and then passes the knowledge to the inference mechanism for reasoning [7]. Whenever the new query comes from the system will learn whether that query is related to the previous query or it generates from the previous query and check how many times users ask the combination of these [7]. We have used the association learning rule mining in the system for making the system exhibit characteristics of being intelligent.

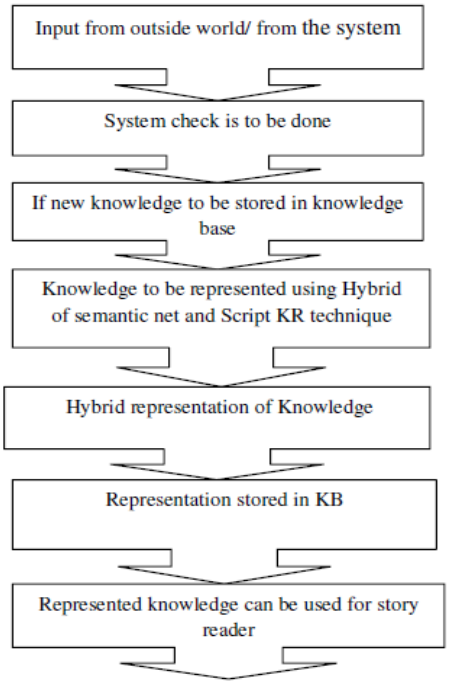


Fig. 6. Methodology of the KB

d) Reasoning Algorithm

Reasoning [24] system is used for getting new fact from the existing knowledge or to draw inference for the situation. The inference can be inductive / deductive. There are many algorithms available for searching the goal and are divided in to two categories i.e. a) uninformed search techniques (depth first and breadth first) and b) heuristic/informed search techniques (best first search, A*, AO*, etc.) in AI. Resolution and chaining (forward and backward) are the known reasoning techniques. Forward chaining refers to deduction whereas the backward chaining refers to induction. An example is considered here for deductive and inductive reasoning. The example of former is "Poonam must be either cooking or washing clothes". If she is not cooking she must be washing clothes. i.e., in case of deductive reasoning the truth of premises must leads to truth of conclusion. The example of inductive i.e., is "the initial failure of machine was caused by some spare part failure". i.e., the truth of premise supports the conclusion without giving exact assurance.

In forward chaining each statement is act as premises and each rule is divided to two parts the left side and the right side. The left side of the rule is being used to match with current condition. If the current condition matches with any or more than one rules on the left side then the right side of the matched rules are applied as the action to be performed. In the same way, system starts from the initial premises and applies forward chaining and moves towards the goal i.e., the conclusion of the given knowledge.

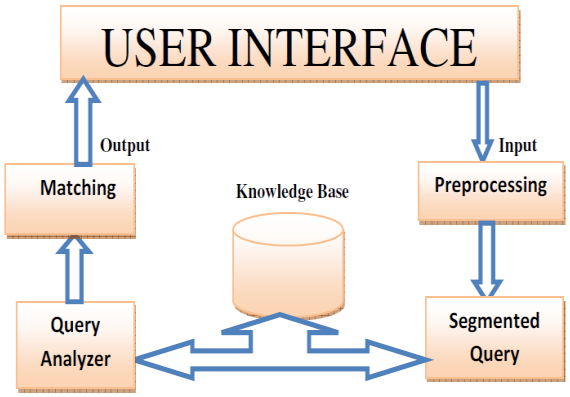


Fig. 7. Architecture of an artificial story reader System as QAS.

The system is able to adequately provide answer for any query related to the input using forward reasoning. Since every problem is unique in nature and has its own complications/ difficulties, the problem domain will require extensive and unique set of knowledge to solve the identified problem. AI requires a collection of knowledge and control mechanism to resolve a specific problem in a systematic fashion to yield the result to satisfaction.

Algorithm used by the system for query applier is given below.

Algorithm Forward (KR, KB, K): returns a substitute that was not found in K.

1. Repeat until KB is empty or NEW is not found.
2. Initialize NEW
3. NEW ← { Ɛ }
4. For each statement S in KB do.
5. (S1 ^ S2 ^ S3^--- ⇒ Q) ← (A)
6. Term ← λ
7. For each λ in Q : such that (S1 ^ S2 ^ S3) λ = (S1′ ^ S2′ ^ S3′) λ in story for some S1′, S2′,S3′ --- in KB.
8. Q′ ← (λ ,Q)

If Q′ not in query then

9. Q′ ← NEW
10. Add Q′ to KB otherwise
11. No answer.

The output of the above algorithm is a combination of words.

III. Result And Conclusion

The implementation of system is under processing. Tokenization and tagging of English statement are very important phases of the system.

The system is able to provide answer from a 1000-lines input provided in the form of a short story, write-up, etc.

The proposed reasoning algorithm that uses semantic net and script will be used for hybrid KR system and can be used in daily life activities as the system is capable of representing such knowledge. Combination of declarative and procedural

techniques makes the system interactive and user friendly. This algorithm will be used for declarative as well as procedural knowledge. The reasoning algorithm can be utilized in many applications of AI and robotics. The proposed reasoning algorithm is used to infer the knowledge from the existing knowledge base. In its advanced stages of development, the proposed system can act as an intelligent system like a QAS.

REFERENCES

[1] Benjamin Cummings. Brewster, et al., , Knowledge representation with ontology's: the present and future, IEEE Intelligent Systems, pp. 72-81, 2004

[2] R. Davis, H. Shrobe, and P. Szolovits, "What is a Knowledge Representation?", AI Magazine, 14(1):17- 33, 1993

[3] Poonam Tanwar, Dr. T. V. Prasad and Dr. Kamlesh Datta, "Hybrid Technique for Effective Knowledge Representation", In "Advances in Intelligent Systems and Computing", Springer, Volume 178, 2012, pp 33-43.

[4] E. Rich and K. Knight, "Artificial Intelligence", 2nd Edition, McGraw-Hill, 1991.

[5] John F. Sowa, "Encyclopedia of Artificial Intelligence", Wiley, 2nd edition, 1992.

[6] Brachman R and Levesque H, eds., "Readings in Knowledge Representation", Morgan Kaufman, 1985.

[7] Poonam Tanwar, T. V. Prasad and Kamlesh Datta, "An Effective Knowledge base system Architecture and Issues in Representation Techniques", Int. J. of Advancements in Tech. available at http://ijict.org/,2011.

[8] Poonam Tanwar, T. V. Prasad and Mahendra. S. Aswal, "Comparative Study of Three Declarative Knowledge Representation Techniques", Int. J. on Computer Sc. and Engg, Vol. 2, No. 7, 2010, pp. 2274-2281.

[9] Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2009

[10] Morgenstern, "Knowledge Representation", Columbia University, 1999, available at http://wwwformal.stanford.edu/leora/krcourse/.

[11] Brachman and. Schmolze, "An overview of the KL-ONE Knowledge Representation System," Cognitive Science, Volume 9, Issue 2, Elsevier, pp 171-216, 1985.

[12] Tsai, Aoyama and Chang, "Rapid Prototyping using FRORL Language", Dept of Electrical Engg. and Computer Sc., Univ. of Illinois at Chicago, Chicago, IEEE, 1988.

[13] Tsai, Jaiig and Karen, Schellinger," RT-FRORL: A Formal Requirements Specification Language for Specifying Real-Time Systems", University of Illinois, Chicago, IEEE transaction, 1991.

[14] L William. Kuechler and Lim,Vaishnavi," A Smart Object Approach To Hybrid Knowledge Representation and Reasoning Strategies", Proc. of the 28th Annual Hawaii Int. Conf. on System Sciences, 1995.

[15] Shetty, Pierre Riccio, Quinqueton, "Extended Semantic Network for Knowledge Representation", Information Reuse and Integration, France, "IEEE-IRI, 2009.

[16] Singhe, Madur and Apperuma," Enhanced Frame-based Knowledge Representation for an Intelligent Environment", IEEE., KIMAS, Boston, USA, 2003.

[17] Poonam Tanwar, T.V. Prasad and Kamlesh Datta, "Hybrid Technique for Effective Knowledge Representation and a Comparative Study", Int. J. of Computer Sc. & Engg Survey, Vol.3, No.4, pp 43-57, 2012

[18] Stillings, Luger, "Knowledge Representation", Chapters 4 and 5, 1994, available at www.hbcse.tifr.res.in/jrmcont/notespart1/node28.htm.

[19] Reena T. N. Shetty, Pierre-Michel Riccio and Joël Quinqueton," Hybrid Model for Knowledge Representation", ICHIT '06 Proc. of the 2006 Int. Conf. on Hybrid Info. Tech., Vol. 01, 2006.

[20] Rathke, C., "Object-oriented Programming and Frame-based Knowledge Representation", 5th Int. Conf., Boston, 1993.

[21] Anne R, Diekema, et al., "What do you mean? Finding answer to complex Questions", National Aeronautics and Space Administration, New York.

[22] David Elworthy," Question Answering using a large NLP System", Microsoft Cambridge , Filtering Track, 2001.

[23] Mukul Aggarwal, "Information retrieval and Question answering NLP Approach: An Artificial Intelligence Application", Int. J. of Soft Computing & Engg., June 2011.

[24] James Allen, George Ferguson, Daniel Gildea, Henry Kautz and Lenhart Schubert, "Artificial Intelligence, Natural Language Understanding, and Knowledge Representation and Reasoning", Natural Language Understanding, 2nd ed., Benjamin Cummings, 1994

A Survey of Automated Text Simplification

Matthew Shardlow
Text Mining Group, School of Computer Science
University of Manchester, Manchester, United Kingdom
Email: mshardlow@cs.man.ac.uk

***Abstract*—Text simplification modifies syntax and lexicon to improve the understandability of language for an end user. This survey identifies and classifies simplification research within the period 1998-2013. Simplification can be used for many applications, including: Second language learners, preprocessing in pipelines and assistive technology. There are many approaches to the simplification task, including: lexical, syntactic, statistical machine translation and hybrid techniques. This survey also explores the current challenges which this field faces. Text simplification is a non-trivial task which is rapidly growing into its own field. This survey gives an overview of contemporary research whilst taking into account the history that has brought text simplification to its current state.**

Keywords*—*Text Simplification, Lexical Simplification, Syntactic Simplification

I. Introduction

Text Simplification (TS) is the process of modifying natural language to reduce its complexity and improve both readability and understandability. It may involve modifications to the syntax, the lexicon or both. The automation of this process is a difficult problem which has been explored from many angles since its conception in the nineties [1]–[7]. This survey paper is intended to give an overview of the field of TS in its current state. To the author's knowledge, there is no similar publicly available survey since 2008 [8]. Whereas the previous survey identified eight separate systems, this work has exposed closer to fifty. The recent growth in TS research can be seen in Figure 1 where it is clear that TS is steadily increasing in size as a field. The last few years have seen a growing maturity in the field, marked by an increased use of both external resources [9]–[11] and methods [12]–[14].

Whereas there has been much work in manual TS over the years, especially with a focus on second language learners [15], there is less work in automated simplification. The first effort towards automated simplification is a grammar and style checker developed for writers of simplified English [16]. This was developed at Boeing for the writers of their commercial aircraft manuals, to help them keep in accordance with the ASD-STE100 standard for simplified English[1]. Further work to automate simplification for controlled language was undertaken [17]. This was later extended for the case of general language in the areas of syntactic simplification [3] and lexical simplification [4]. These methods have heavily influenced future efforts to date. Work to improve the preservation of discourse in syntactic simplification [18] and to improve the context–awareness of lexical simplification [12]–[14] has been carried out. Other work has involved applying phrase based statistical machine translation techniques to produce simple English [10], [19], [20].

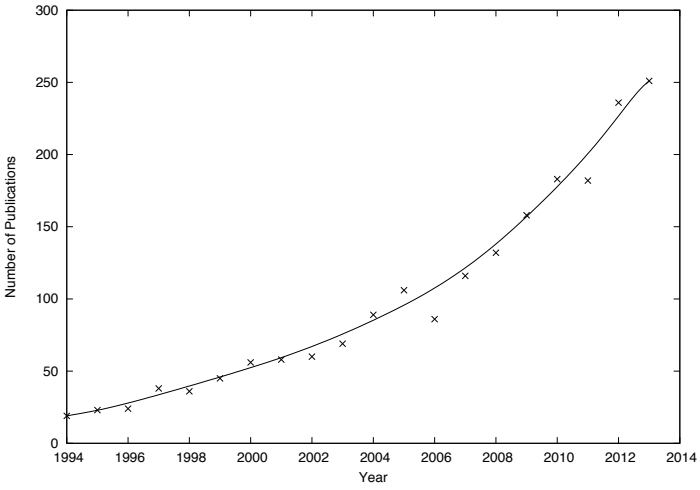


Fig. 1. This graph was produced by polling Google Scholar with the search query: 'Text Simplification' OR 'Lexical Simplification' OR 'Syntactic Simplification'. It shows the sustained growth in TS and associated sub-fields between 1994 and 2013.

TS is within the field of natural language processing. Within this field it is very similar to other techniques such as machine translation, monolingual text-to-text generation, text summarisation and paraphrase generation. These fields all draw on each other for techniques and resources and many techniques within TS come from these other fields [19], [21]. TS is different to text summarisation as the focus of text summarisation is to reduce the length and content of input. Whilst simplified texts are typically shorter [22], this is not necessarily the case and simplification may result in longer output — especially when generating explanations [23]. Summarisation also aims at reducing content — removing that which may be less important or redundant. This is typically not explored within simplification, where all the content is usually kept. Some efforts have explored the use of simplification alongside summarisation systems [24]–[28]. Here, TS is used to improve the readability of the final summary.

When talking about TS the words *simple* and *complex* are often used in relation to each other as shown in Table I. For example, in a parallel corpus of simplified English and regular English, the former will be called *simple* and the latter *complex*. In a corpus of technical English and regular English, the former will be called *complex* and the latter *simple*. This shows that simplicity and complexity are relative to each other and should be used with care. When creating *simple* text, we actually intend to create text which is more *simple* (and so

[1]http://www.asd-ste100.org/

TABLE I. SIMPLICITY IS RELATIVE TO COMPARISON

	Simple Text	Lay Text	Technical Text
Simple vs. Lay	Broken Arm. > Fractured Arm. (is simpler than)		
Lay vs. Technical		Fractured Arm. > Fractured Tibia. (is simpler than)	

less *complex*) than it originally was. Two other important terms to define are *readability* and *understandability*. At first, these may seem like the same things, however they may be measured independently depending on the context of an application. *Readability* defines how easy to read a text may be. This is typically governed by factors such as the complexity of grammar, length of sentences and familiarity with the vocabulary. *Understandability* is the amount of information a user may gain from a piece of text. This can be affected by factors such as the user's familiarity with the source's vocabulary, their understanding of key concepts or the time and care taken to read the text. It may be the case that a text has high *readability*, but low *understandability*. For example: trying to read a well written scientific article with no scientific training. It may also be possible that a text has low *readability*, but is still *understandable*. For example: an author who communicates a simple point uses misleading grammatical structures. *Readability* and *understandability* are related and a text which is easier to read is likely to be more understandable, as the reader will find it easier to take the time to look over the difficult concepts. Similarly, a text which is easily understandable will encourage the reader to keep reading, even through difficult readability.

Simplicity is intuitively obvious, yet hard to define. Typical measures take into account factors such as sentence length [29], syllable count [30] and other surface text factors. Whilst these give a good estimate, they are not always accurate. For example, take the case of sentence length. One long sentence may use many complex terms and anaphora (words which refer to a previous entity: he, she, it, etc.). A simplified version of this would be lexically longer, but may be more explicative. In the case of explanation generation, complex words are appended with short definitions, increasing sentence length. Automatic heuristic measures will judge these sentences which have been simplified to be more complex. The final text may be longer, however it is also easier to understand and therefore simpler.

Different forms of simplification will address different needs. No two users are exactly the same and what one finds easy, another may find difficult. This is true both at the level of different user groups (the low literacy user requires different simplifications to the second language learner), but also within user groups. Factors such as dialect, colloquialisms and familiarity with vocabulary and syntax can all affect the user's understanding of a text. This means that simplification is best done at a general level. Text which is made very simple for one user may be more complex for another. However text which is made slightly simpler for one user will generally be easier for most other users.

This still leaves the question of how to measure simplicity. Automatic measures are ineffective [31]. In fact, they may even be detrimental to the simplification process. For example, if a measure favours short sentences and the aim of simplification is to get the best score with that measure, we could easily succeed by reducing our text to a series of two or three word stubs. This would be much more difficult to read and understand, yet score highly.

Although many efforts have been made towards TS techniques over the past two decades, few have been used in production. Those used in production are generally developed with a focus as an aid to the user in translating their text to simplified language [16], [32]. A governing factor in the low take-up of TS systems is inaccuracy. In some natural language processing applications, a low accuracy may be acceptable as the application is still usable. For example, in information retrieval, even if a system has a moderate accuracy, it will still enable the user to find some portion of the documents they were looking for. Without the information retrieval system, the user would not have been able to find the documents as easily. However, this does not transfer in the same way to TS. If a system is not accurate, then the resultant text will not make sense. If the text is not understandable, then it will definitely not be more simple than the original. If a user is routinely presented with inaccurate simplifications, then they will not find it helpful. Assistive technology must be accurately assistive, otherwise the user will be more confused and less able to interact with the text than in its original "more complex" form.

TS is a largely unsolved task. Whereas many areas of natural language processing and computer science have a flagship system, method or technique, TS has many varied approaches (as outlined in Section II). Whilst these techniques employ differing methodologies and may have differing outputs, their purpose is always to simplify text. The field is fast moving and research into new areas is regularly produced. Further, more and more people are becoming interested in TS with an increased number of projects and publications year on year, as shown in Figure 1. Whilst many techniques have been implemented, there is still much work to be done in comparing, evaluating and refining these.

Text is a fundamental part of our daily interaction with the information world. If text is simplified for an end user, then this may improve their experience and quality of life. Whether we are reading the newspaper, checking emails or following instructions, it is highly important to be able to understand the text used to convey this information. TS can be applied to reduce the complexity of information and increase a user's understanding of the text they encounter in their day to day lives. This has great advantages for both readers and authors. The reader gains a better understanding of the world around them and authors can ensure their written material will be understandable by those recipients with a low reading level.

The related fields of machine translation and text summarisation allow for the crossover and sharing of techniques. For example, corpus alignment techniques have been borrowed from summarisation [33], [34] and statistical machine translation techniques have been used [19], [35], along with their evaluation methods [36]. This crossover means that, as these fields progress, there will be new techniques available for the task of simplification. As techniques are developed in the context of TS, they will also be useful in the context of other related domains.

The need for simplified English in particular is evidenced by the popularity of the Simple English Wikipedia project (an alternative to English Wikipedia), which provides simplified versions of Wikipedia articles. There are over 88,000 articles which have been hand written in Simple English for this project. Many groups with low levels of English benefit. The size of Simple Wikipedia indicates the need for simple English, however the process of hand crafting these articles is time consuming. Improvements in automating simplification would help to address this need.

II. Approaches

TS has been carried out in a number of different ways. Many systems use a combination of approaches to simplify text in different manners. These different methods of TS are largely independent and methodologically distinct of each other. In this section, we observe the development of methods from: lexical and syntactic simplification, explanation generation, statistical machine translation and TS techniques in languages other than English.

A. Lexical Approaches

Lexical simplification is the task of identifying and replacing complex words with simpler substitutes. This involves no attempt to simplify the grammar of a text but instead focusses on simplifying complex aspects of vocabulary. An overview of research papers in lexical simplification is presented in Table II. Lexical simplification may be formulated as a phrase based substitution system, which takes limited syntactic information into account. There are typically 4 steps to lexical simplification as shown in Figure 2. Firstly, the complex terms in a document must be identified. Secondly, a list of substitutions must be generated for each one. Thirdly, those substitutions should be refined to retain those which make sense in the given context. Finally, the remaining substitutions must be ranked in their order of simplicity. The most simple synonym is used as a replacement for the original word. Systems have made differing variations on this theme with many approaches missing out the word sense disambiguation step.

In the first notable work in automated lexical simplification [4], the authors rank synonyms from the semantic thesaurus WordNet [49] using Kučera-Francis frequency [50] to identify the most common synonym. This work has heavily influenced lexical simplification systems since [12]–[14], [34], [37], [51], providing a framework with many avenues to explore and build upon. Recently, work has also focussed on the simplification of numerical expressions for improved reader comprehension [52], [53].

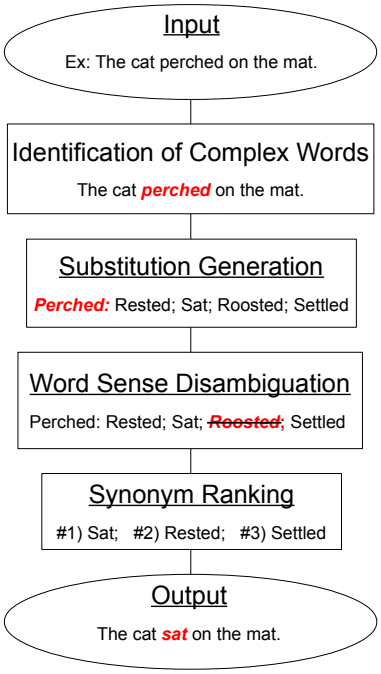


Fig. 2. The lexical simplification pipeline. Many simplifications will be made in a document concurrently. In the worked example the word 'perched' is transformed to sat. 'Roosted' is eliminated during the word sense disambiguation step as this does not fit in the context of 'cat'.

One area for improvement is the method of substitution ranking. Kučera-Francis frequency is the counts of words from the Brown corpus, which consists of just over one million words from 50 sources which are intended to be representative of the English language. Modern technology allows for frequency counts of much larger corpora to be carried out [54], [55]. Larger corpora are naturally better estimators of the true frequency counts of a language.

One of the major stumbling blocks with primitive lexical substitution systems is a loss of meaning due to word sense ambiguity. This occurs when a word has multiple meanings and it is difficult to distinguish which is correct. Different meanings will have different relevant substitutions and so replacing a word with a candidate substitution from the wrong word sense can have disastrous results for the cohesion of the resultant sentence. Early systems [4] did not take this into account, at the expense of their accuracy. Word sense disambiguation may be used to determine the most likely word sense and limit the potential synonyms to those which will maintain coherence.

Word sense disambiguation has been applied to lexical simplification in a number of different ways. These usually involve taking a standard lexical substitution system and applying a word sense disambiguation algorithm at some point. One such system is the latent words language model (LWLM) [56], which is applied to lexical simplification during the substitution generation step. The LWLM is used to generate a set of words which are semantically related to the original word. These are then compared against the substitutions returned by WordNet to remove any antonyms found by the LWLM. WordNet is useful for word sense disambiguation as it gathers words according to their semantic similarities into a group

TABLE II. RESEARCH EFFORTS IN LEXICAL SIMPLIFICATION ORDERED BY YEAR. LATER SYSTEMS ARE TYPICALLY MORE SOPHISTICATED. RECENT YEARS HAVE SEEN THIS AREA GATHERING MOMENTUM.

Year	Title	Notes
1998	The Use of a Psycholinguistic Database in the Simplification of Text for Aphasic Readers (PSET) [4]	Seminal work on lexical simplification.
2003	Text Simplification for Reading Assistance: A Project Note (KURA) [37]	Paraphrasing for deaf Japanese students in an educational setting.
2006	Helping Aphasic People Process Online Information (HAPPI) [38]	An update of PSET project for Web deployment.
2007	Mining a Lexicon of Technical Terms and Lay Equivalents [39]	Corpus alignment for paraphrasing.
2009	FACILITA: Reading Assistance for Low-literacy Readers (PorSimples) [40]	Designed for Brazilian Portuguese readers.
2009	Extracting Lay Paraphrases of Specialized Expressions from Monolingual Comparable Medical Corpora [41]	Paraphrasing medical corpora.
2010	Lexical Simplification [13]	Applying word sense disambiguation during the synonym generation phase.
2010	For the Sake of Simplicity: Unsupervised Extraction of Lexical Simplifications from Wikipedia [9]	Paraphrasing.
2011	Putting It Simply: a Context-aware Approach to Lexical Simplification [12] (SIMPLEXT)	A word sense disambiguation approach to lexical simplification.
2012	Can Spanish Be Simpler? LexSiS: Lexical Simplification for Spanish [42]	Spanish lexical simplification. c.f. [43], [44]
2012	English Lexical Simplification (SemEval Task 1) [45]	The project description for the SemEval 2012 task on lexical simplification.
2012	WordNet-based Lexical Simplification of a Document [46]	using WordNet hypernymy to perform substitutions.
2012	Automatic Text Simplification via Synonym Replacement [47]	Masters thesis focussing on the challenges of lexical simplification in Swedish.
2013	User Evaluation of the Effects of a Text Simplification Algorithm Using Term Familiarity on Perception, Understanding, Learning, and Information Retention [48]	Semi-automated lexical simplification for medical literature.

called a "synset". One particular use of WordNet [46] develops a tree of simplification relationships based on WordNet hypernym relations. This tree is used to reduce the size of the vocabulary in a document. Word sense disambiguation is carried out to place content words into their correct WordNet synset. Simplification may then be carried out by looking at the relevant node in the tree. Word sense disambiguation is also carried out by the use of context vectors [12], [42]. In this method, a large amount of information is collected on the surrounding context of each word and is used to build a vector of the likely co-occurring words. Vector similarity measures are then used to decide which word is the most likely candidate for substitution in any given context. These methods show the diversity of word sense disambiguation as applied to lexical simplification.

Other work has attempted to improve lexical simplification by improving the frequency metrics which are used. Frequent words have been shown to increase a text's readability [57]. Simple Wikipedia has been shown to be more useful than English Wikipedia as a method for frequency counting [58]. N-Grams have shown some use in providing more context to the frequency counts, with higher order n-grams giving improved counts [59]. However, the most effective method has so far proven to be the usage of a very large initial data-set [58], [59]. Namely, the Google Web 1T [55].

As well as performing substitutions at the single word level, lexical substitution may also be carried out at the phrase level, which requires some knowledge of how words cluster into individual phrases and how these can be recognised and substituted. A phrase may be replaced by a single word which conveys the same sentiment or by another phrase which uses simpler language. This may be done by comparing revisions in the edit histories of Simple Wikipedia [9] or by comparing technical documents with simplified counterparts [39]. A corpus which can be used to identify simplifications made by a human editor is required. Phrase based simplification is

similar to the task of paraphrasing [41], [60], where phrases with high semantic similarity are aligned for use in tasks such as question answering [61] or the automatic evaluation of machine translation [62]. Techniques could be drawn from this area to improve the work in lexical simplification. Two advantages are as follows: Firstly, it allows some rudimentary syntactic simplification to be carried out, altering the structure within a phrase to make it more readable. Secondly, it allows more diversity in the range of simplifications which can be made. It may be the case that simplifying a single word which is part of a complex phrase is actually detrimental to the understanding of that phrase, whereas simplifying the whole phrase itself is helpful.

A recent important development in the field of lexical simplification is the lexical substitution task from SemEval 2012 [45]. Participants designed a system to rank words in terms of their simplicity. The words were given as valid replacements for a single annotated word in a sentence. Many such sentences were provided and systems were able to train and test on sample data before being deployed for the final testing data. The corpus was developed by crowd sourcing through Amazon's Mechanical Turk[2]. Annotators were asked to rank the substitutions in order of their simplicity. These rankings were then combined to form one final ranking.

The SemEval task isolates the synonym ranking problem within lexical simplification where the aim is to find the easiest synonym. Systems do not have to focus on other distractions, such as identifying complex words or synonym generation, but can focus solely on ranking. Several systems were developed to produce these rankings and the techniques used considered a variety of methods such as: language models for word context [63]–[65], decompositional semantics [66] and machine learning techniques [65], [67]. A comprehensive overview and comparison of these is given in the task description [45]. The SemEval task benefits TS in two separate ways: firstly, it has promoted the field and specifically the area of lexical simplification. Hopefully, interest will be generated and more time and resources will be channeled into TS. Secondly, it has provided an evaluation of different methods for synonym ranking. This should drive research forward as new systems will have both a reasonable baseline and evaluation method to compare against.

B. Syntactic Approaches

Syntactic simplification is the technique of identifying grammatical complexities in a text and rewriting these into simpler structures. There are many types of syntactic complexity which this may apply to: Long sentences may be split into their component clauses; Sentences which use the passive voice may be rewritten and anaphora may be resolved. Poorly written texts are very difficult to engage with. Readers may struggle to follow the text, lose interest at some point in a sentence and eventually give up trying. In the case of people with cognitive impairments such as aphasia, some grammar structures may even cause a loss of meaning. Patients may not be able to distinguish between subject and object when the passive voice is used. For example, the passive voice sentence: "the boy was kicked by the girl" may appear to read as: "the boy kicked the girl" for someone with aphasia. A list of research in syntactic simplification is presented in Table III.

[2]www.mturk.com

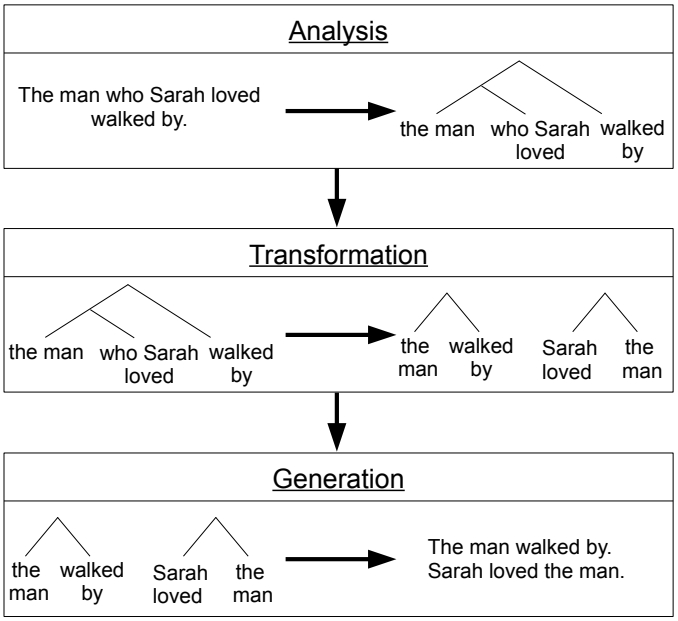


Fig. 3. The syntactic simplification pipeline, with worked example. Predetermined rewrite rules govern the simplifications that occur during the transformation step. The generation step is important to ensure the cohesion of the resultant text.

Work on syntactic simplification began with a system for the automatic creation of rewrite rules for simplifying text [3]. This system takes annotated corpora and learns rules for domain specific sentence simplification. The main purpose is as a preprocessing step to improve other natural language applications. Later work [68], [75], [77], [79] focussed on applying this syntactic simplification as an assistive technology. Improvements to the discourse structure were made to ensure that clauses of sentences appeared in the correct order [18]. More recent work has focussed on applying syntactic simplification as a preprocessing tool for named entity recognition in the biomedical domain [5], [26]. There have also been efforts to apply this technique for languages other than English [51], [69], [71]–[73].

Syntactic simplification is typically done in three phases as shown in Figure 3. Firstly, the text is analysed to identify its structure and parse tree. This may be done at varying granularity, but has been shown to work at a rather coarse level. At this level, words and phrases are grouped together into 'super-tags' which represent a chunk of the underlying sentence. These super-tags can be joined together with conventional grammar rules to provide a structured version of the text. During the analysis phase, the complexity of a sentence is determined to decide whether it will require simplification. This may be done by automatically matching rules, but has also been done using a support vector machine binary classifier [80]. The second phase is transformation, in which modifications are made to the parse tree according to a set of rewrite rules. These rewrite rules perform the simplification operations such as sentence splitting [68], clause

TABLE III. THE STATE OF SYNTACTIC SIMPLIFICATION RESEARCH ORDERED BY YEAR. RECENT EFFORTS HAVE PARTICULARLY SEEN THIS AS APPLIED TO LANGUAGES OTHER THAN ENGLISH.

Year	Title	Notes
1997	Automatic Induction of Rules for Text Simplification [3]	Seminal work in field.
1998	Practical Simplification of English Newspaper Text to Assist Aphasic Readers (PSET) [68]	Shortened sentences for aphasic users.
2004	Automatic Sentence Simplification for Subtitling in Dutch and English [69]	Dutch language simplification
2004	Text Simplification for Information-seeking Applications [6]	Introduce the notion of Easy Access Sentences.
2006	Syntactic Simplification and Text Cohesion [18]	Maintaining discourse when performing syntactic simplification
2009	Sentence Simplification Aids Protein-protein Interaction Extraction [70]	Preprocessing for biomedical interaction recognition.
2010	A Semantic and Syntactic Text Simplification Tool for Health Content [23]	Long sentences split after explanation generation.
2010	Simplifica: a Tool for Authoring Simplified Texts in Brazilian Portuguese Guided by Readability Assessments (PorSimples) [32]	An authoring tool which provides text simplification techniques whilst writing a document.
2012	Acquisition of Syntactic Simplification Rules for French [71]	A comprehensive list of rules for simplifying the French language.
2012	Sentence Splitting for Vietnamese-English Machine Translation [72]	Vietnamese language splitting to improve machine translation.
2012	Transforming Complex Sentences using Dependency Trees for Automatic Text Simplification in Basque [73]	Basque language syntactic simplification.
2012	Enhancing Multi-document Summaries with Sentence Simplification [26]	Syntactic simplification as a preprocessing aid.
2013	ERNESTA: A Sentence Simplification Tool for Children's Stories in Italian [74]	Italian Syntactic Simplification
2013	Enhancing Readability of Web Documents by Text Augmentation for Deaf People [75]	Simplification of Korean for deaf readers.
2013	Sentence Simplification as Tree Transduction [76]	Direct manipulation of parse trees.
2013	Simple, Readable Sub-sentences [77]	Removing unnecessary parts of sentence
2013	Corpus-based Sentence Deletion and Split Decisions for Spanish Text Simplification [78]	Spanish syntactic simplification.

rearrangement [18] and clause dropping [74], [78]. Although techniques for automatically inducing these rules exist [3], most other systems implementing syntactic simplification use hand written rewrite rules. Two reasons for this are the removal of the need for annotated corpora and the improved accuracy of the final rules. After transformation, a regeneration phase may also be carried out, during which further modifications are made to the text to improve cohesion, relevance and readability.

Syntactic simplification is an essential component to any working TS system and has been implemented in both PSET [4] and PorSimples [51] which both seek to provide ubiquitous TS as an assistive technology. It has been particularly useful outside this application and has been implemented for improving the accuracy of other natural language techniques with significant success. Syntactic simplification will be incorporated into future TS systems, as it has the ability to reduce grammatical complexities in a way which is not possible with other techniques. Creation and validation of the rewrite rules is a difficult process and one aspect of further work may concentrate on new techniques to automatically discover these.

C. Explanation Generation

Explanation generation is the technique of taking a difficult concept in a text and augmenting it with extra information, which puts it into context and improves user understanding. Table IV lists the research in this area. It has been shown that in some cases, this is more appropriate than lexical simplification [83]. A specific example can be taken from

TABLE IV. RESEARCH INTO EXPLANATION GENERATION, ORDERED BY YEAR.

Year	Title	Notes
2006	SIMTEXT Text Simplification of Medical Literature [81]	Dictionary definitions appended for some terms.
2009	FACILITA: Reading Assistance for Low-literacy Readers (PorSimples) [40]	References to Wikipedia articles for difficult terms.
2010	A Semantic and Syntactic Text Simplification Tool for Health Content [23]	Long sentences split after explanation generation.
2012	Sense-specific Lexical Information for Reading Assistance [82]	Lexical elaboration for the education of second language learners.

Health Informatics, where explanations are generated for terms in health literature [23], [81]. These are categorised by their semantic type (disease name, anatomical structure, device, etc.) Explanations are then generated by finding an easier term and adding in a short connecting phrase to explain the more complex term:

"Pulmonary atresia *(a type of birth defect)*"[3]

'Pulmonary atresia' is found to be semantically related to 'birth defect' and the connecting phrase 'a type of' is added to maintain cohesion. Five semantic types are identified and a medical thesaurus is employed for identifying valid substitutions. This is highly specific to the medical terminology in question. The semantic types were discovered by manually analysing simplified literature and so applications to the general case would require much analytical work and many more categories to be discovered. Whilst analysis could be automated, the accuracy would then suffer. This technique could also be used in another equally specific technical domain.

A more general form of simplification is carried out as part of the PorSimples project [51]. The application 'Educational FACILITA' [84], provides a browser plug-in which can simplify Web content for users. Named entities are recognised and annotated with short explanatory extracts from Wikipedia articles. These allow the user to learn more about the difficult concepts in a text. This information is presented to the user in a separate text box at their request. Lexical simplification is also carried out to improve the text's overall readability. The largest challenge lies in the named entity labelling task. Here, words must be matched to their semantic concepts. This is a difficult task which requires word sense disambiguation and some mapping between the concepts and their explanations.

More recently, this has been applied to the case of second language learners [82]. Here, the learner has the opportunity to highlight words which they find difficult and see a dictionary entry for that word. Word sense disambiguation (as discussed above in Section II-A) is carried out to ensure that only the correct sense of the word is presented to the user. This is shown to increase the language learner's reading comprehension for the explained words.

In its present form, explanation generation has particular potential for users with some understanding who wish to learn more about a text. By providing explanations alongside difficult terms, the user is able to better understand the concept and will hopefully not require the explanation next time they encounter the complexity. Explanation generation is not confined to presentation alongside the complex terms however and may also be done to replace the original word. A semantically simple phrase which explains the original term could be used as its replacement. Due to the potentially complex levels of processing involved, this is a technique which is prone to error. If errors occur and are left undetected and unresolved, then they may result in the final text becoming misleading and unhelpful to an end user, which should naturally be avoided wherever possible. This technique may also be useful when deployed alongside lexical [84] or syntactic simplification [23]. The explanations which are generated may add to the structural complexity of the text, resulting in diminished readability. Any steps to increase the readability will help the reader to interact with the text.

[3]From [23]. Generated explanation in italics

D. Statistical Machine Translation

Automated machine translation is an established technique in natural language processing, for a comprehensive review see [88]. It involves automatic techniques to convert the lexicon and syntax of one language to that of another, resulting in translated text. Its application to TS involves casting our problem as a case of monolingual text-to-text generation. Table V gives the research in this field to date. We consider our translation task as that of converting from the source language of complex English to the target of simple English. Each has its own unique syntax and lexicon and is sufficiently distinct to permit the use of machine translation techniques. Recent research (as described below) in machine translation has focussed on phrase based statistical techniques. These learn valid translations from large aligned bilingual corpora and are then able to apply these to novel texts. This task is made easier as the source and target languages are very similar, and so few changes are necessary. It is this type of machine translation that has been applied to TS.

Work to perform TS by statistical machine translation has been performed for English [10], [19], [20], Brazilian Portuguese [35] and has been proposed for German [87]. Practically, systems often use and modify a standard statistical machine translation tool such as Moses [89]. A difficult task can be finding aligned sentences in complex and simple language. This has been done by manual creation [35] and by mining English and Simple Wikipedia [19] using techniques from monolingual corpus alignment [90].

Using this corpus, Moses has been applied to the TS task for English [10]. Moses was augmented with a phrase deletion module which removed unnecessary parts of the complex

TABLE V. PAPERS PRESENTING TS BY STATISTICAL MACHINE TRANSLATION ORDERED BY YEAR.

Year	Title	Notes
2010	Translating from Complex to Simplified Sentences (PorSimples) [35]	Part of the PorSimples project for TS in Brazilian Portuguese
2010	A Monolingual Tree-based Translation Model for Sentence Simplification [19]	Tree based model, produced the PWKP dataset of aligned complex-simple sentences from Wikipedia.
2011	Learning to Simplify Sentences using Wikipedia [10]	Improves previous work.
2012	Sentence Simplification by Monolingual Machine Translation [20]	Further improves on previous work.
2012	A Simplification Translation Restoration Framework for Cross-domain SMT Applications [85]	Chinese – English. Simplification as processing aid.
2013	Statistical Machine Translation with Readability Constraints [86]	English – Swedish. Simplification for improved readability.
2013	Building a German/Simple German Parallel Corpus for Automatic Text Simplification [87]	A corpus for the production of German monolingual statistical machine translation simplification.

source text. The evaluation used BLEU [91], a standard measure in machine translation. Recent research [20] has used human judges to evaluate the quality of the simplified text against a lexical substitution baseline, something which has not been done before. The use of human judges is a valuable method for evaluation in TS.

A recent development has been simplification in more traditional bilingual statistical machine translation, which has occurred for translating English to Chinese [85] and Swedish to English [86]. Simplification aids readability in the target language, making it useful for language learners. It is also useful to transform source and target texts to a common format to improve their alignment, thus improving translation accuracy. In the example of English to Chinese translation, the final text is restored to it's original level of complexity, the simplification is only required for improving the quality of the translation.

As new techniques and evaluation methods are developed for machine translation, they will be directly applicable to this task. Simplification through monolingual machine translation gives a form of simplified text which appears to reflect human simplified text. This may be useful when simplifying for different domains as the types of simplification are automatically learnt by the algorithm. Statistical machine translation is a technique with applications in real world systems. However, the nature of a statistical technique is that it will not work perfectly in every single case. Every statistical technique has a number of false positives (simplification operations made in error) and false negatives (simplifications which should have been made). Whilst the aim is to reduce these at the same time as improving the levels of true positives and negatives, there will always be some errors that creep in during the learning process. As discussed previously (see Section I), the introduction of errors results in a diminished understandability and increased text complexity — the opposite to the desired outcome. This highlights the importance of accuracy and output validation in TS.

E. Non-English Approaches

As with many natural language processing applications, the majority of TS research is conducted solely for the English language. However, TS is also applied across many different languages as shown in Table VI. The KURA project [37] worked on Japanese language simplification for deaf students and introduced the concept of phrase based simplification identifying and simplifying complex terms. Similarly, the PorSimples project has contributed much to the wider field of TS. This is undoubtedly the largest TS project to date with 3 main systems and many types of simplification investigated. The Simplext project is an ongoing project, currently in the process of developing simplification tools and resources for Spanish. It has particularly focussed on the application of simplification for dyslexic readers.

Most projects do not focus on introducing new techniques for TS, but instead focus on implementing existing techniques in their own language. This is an interesting challenge, as language specific characteristics make it non-trivial to re-implement existing techniques. The main barrier is usually in discovering appropriate resources for the language. For lexical simplification, an extensive word frequency list and some electronic thesaurus is usually employed. If no such word frequency list exists, this may be easily calculated from a count of a sufficiently large corpus (such as pages from Wikipedia). Syntactic simplification typically requires more work to be done. The differences between simplified text and complex text in the language must be analysed to discover language specific simplification rules. These will typically not be transferable between languages due to differing grammar structures. Some constructs such as passive voice and WH-phrases may be common points of confusion across languages and so research may be aided by identifying these known complexities. Techniques to learn these automatically [3] and statistical machine translation may be of use here.

It can be seen from Table VI that recent times have seen a proliferation in TS techniques in languages that are not English. Of the fourteen systems presented, eight have publications in 2012-13. This may be in part due to projects

TABLE VI. A TABLE OF TS IN DIFFERENT LANGUAGES. SS = SYNTACTIC SIMPLIFICATION. LS = LEXICAL SIMPLIFICATION. EG = EXPLANATION GENERATION. PBMT = PHRASE BASED MACHINE TRANSLATION. INFORMATION IS OMITTED WHERE UNAVAILABLE

Year	Language	Methodology	Notes
2003	Japanese (KURA) [37]	LS	No continuing work evident
2004	Dutch [69]	SS	Completed study
2007–10	Portuguese (PorSimples) [51]	SS, LS, EG	Completed study
2010–2013	Spanish (Simplext) [42]	LS	Ongoing work
2011	Italian (Read-it) [92]	LS, SS	No continuing work evident
2012	French [71]	SS	Present a set of syntactic rules
2012	Bulgarian (FIRST) [93]		Preliminary study
2012	Danish (DSIM) [94]	PBMT	Aligned corpus for training
2012	Swedish [47]	LS	Masters Thesis
2012	Vietnamese [72]	SS	Preprocessing for Machine Translation
2013	Basque [73]	SS	Preliminary study
2013	Italian [74] (ERNESTA)	SS	Simplification of children's stories.
2013	Korean [75]	SS	Simplification for sign language users
2013	German [87]	PBMT	Aligned corpus for training

such as PorSimples and Simplext publicising TS as a research field, especially for non-English natural language processing research.

III. RESEARCH CHALLENGES

Throughout this survey, many open areas have been identified and this Section will gather these together and suggest future directions for research. These directions have been grouped into three categories: resources, systems and techniques. Section III-A describes the need for novel evaluation methods and corpora for the further development of existing TS systems. Section III-B outlines the need for TS systems and some methods for the deployment of these. Section III-C explains the need for the development of new algorithms in the field.

A. Resources

Resources are the foundation upon which a system is built. TS has seen many different approaches to the task of providing resources such as evaluation methods and corpora. These have often been done with little consideration to prior techniques and so one general aspect of future work is the comparison and evaluation of potential resources.

Current techniques for automatically evaluating readability are of limited use in TS research. A strong contribution to the field would be an automatic evaluation measure which reliably reported the effects of TS. Some progress has been made [95]. However, this is only useful for the highly specific task of ordering synonyms in terms of their complexity. This is very useful when evaluating a system designed for the specific task, but not as useful for the general task of TS. An evaluation method is needed which has the generality of a readability formula [29], [30], [96] but with the specificity and speed of an automated measure [95].

Manual techniques for the evaluation of automatic TS may also be investigated and developed. Whilst automated techniques give some impression as to the efficacy of a system, they are a step removed from the actual intended audience and so will never be as accurate as direct user evaluation. Many authors have used some manual evaluation for their results [4], [23], [31], [37] and research should aim towards this, especially when deploying a TS system for a specific user group. Experiments to determine the best manual methods of evaluation may also take place.

In addition to research on the evaluation methods, the development of new corpora is equally paramount to the progression of the field. As there are different approaches to TS (see Section II), different types of corpora are necessary. Simplification is inherently difficult to evaluate as there is no obvious correct answer. This means that a corpus cannot be in the standard format of a set of problems labelled with their solutions. Instead, more abstract corpora must be developed to address specific evaluation needs within the TS domain. As these are developed, evaluation methods will be developed alongside them. Corpora which draw on human annotation and where possible the input of the eventual users of a TS system will be more effective than those that do not.

One promising method for corpus development and evaluation comes from the field of statistical machine translation. Some authors have formulated TS as a monolingual translation problem [10], [19], [20]. This creates the possibility of using machine translation evaluation methods such as BLEU [91] and NIST [36]. These techniques compare a given translation

with a reference translation and report an accuracy measure based on the co-occurring words across the two translations. These may also be applicable to the wider scope of TS where sample simplifications could be compared to one or many reference texts. As machine translation evaluation techniques are advanced, the benefits may also be reaped by the text simplification community.

B. Systems

Another research challenge is the development of TS applications. These will exist as a layer of assistive technology upon information gathering systems. There are two clear options for the development of publicly available TS systems, as outlined below.

Firstly, TS can be applied at the user's level. In this model, the user receives some complex text which they automatically simplify by some means. This could take on the form of a Web browser plug-in which allows the user to select and simplify text (similar to the the FACILITA project for Brazilian Portuguese [40]). This could also take on the form of an application which allows the user to identify text and simplify. Some users may not even require the choice to simplify text. For example, in the context of browsing the Internet, some users may find the complex text which is presented to them at first distracting, demoralising or off-putting. Here, it may be helpful to automatically reduce the complexity of any text on a webpage before presenting it to a user.

Secondly, TS may be applied by the author to a text he is creating [97]. In this model, the author may write a document and then use automatic techniques to identify any complexities and to automatically simplify or receive suggestions as to simplifications he may apply. The main advantage is that the author can check the quality of simplifications before the text is presented to a user. Grammaticality, cohesion and intended meaning are definitely preserved, whilst understandability and readability are increased. This is useful in many different applications where text is being written for audiences who may not necessarily understand the final product. The research challenge here is to develop helpful ways of doing this which allow an author to target his text to many levels of understanding.

TS is currently not a commercialised application. This may be in part due to low accuracy in test systems and the youth of the field. As work is done to increase the accuracy of TS systems, they will become more commercially viable. TS is a useful product which can be packaged and sold in the form of software and Web services. As an industry develops around TS, this will create interest in the area which will drive the field to further developments.

C. Techniques

The identification and evaluation of new techniques is paramount to the progression of the field, as the potential solution space for TS is currently sparsely explored. This is mainly because previous projects have been limited by the resources available. As more TS research applications are developed, a few underexplored areas for focus are as follows. These are not intended as an exhaustive list of all the potential future work in TS, but instead to highlight some areas which may be of future interest. These have all been explored initially and references are provided as appropriate.

Firstly, word sense disambiguation is highly important for lexical simplification. Initial work ignored ambiguity in the hope that complex words would belong to only one potential sense. This has not been the case and word sense errors (where a synonym with a drastically different meaning is selected) are a common problem among lexical substitution systems. Some work has previously addressed this [12]–[14], however future efforts must focus on incorporating state of the art word sense disambiguation techniques and adapting these for best use within the TS context. This may be implemented at the synonym ranking step of lexical substitution to combine the simplicity score with a 'relevance' score produced by a disambiguation system. Words which are of low relevance in a context will make the text less understandable.

Secondly, work should be undertaken to improve techniques for identifying candidates for simplification within a text. Whilst there has been plenty of work into readability measures, little has been transferred to a TS setting, although exceptions do exist [21], [80]. Machine learning techniques hold some promise and should be investigated further. The existing techniques are for sentence level simplification and further work could focus on candidate identification at the lexical level. This would involve looking at features of given words and developing some classification system to identify those of sufficient complexity to require simplification.

IV. Conclusion

TS is a domain which has emerged as a reaction to difficult texts. This has occurred for different applications such as preprocessing for machine translation [72] and assistive technology for people with Aphasia [4]. These applications promise to reduce the complexity of text whilst improving readability and understandability. This is a highly useful task and is highly applicable in many settings such as second language learners and lay readers of technical documents. TS is not solely confined to the reader, it may also be applied by the author to a text in order to ensure his point is clearly communicated, or even in a natural language processing pipeline to improve the performance of later components.

There are also many approaches to the task. Some focus on the lexical level, replacing complex words with simpler synonyms. Some modify the syntax of a text to remove complex grammatical structures. Yet others perform phrase based machine translation in an effort to automatically learn valid methods of simplification. The field is currently seeing a wave of growth with many new research projects and new approaches being developed. As the field progresses, more techniques will become available and TS will be widely distributed.

TS is on its way to becoming a household application. As it does so, it is likely that people will often not even know they are benefitting from it. Campaigns for simplified English have existed for many years. TS offers an answer.

References

[1] S. Crossley, D. Allen, and D. McNamara, "Text simplification and comprehensible input: A case for an intuitive approach," *Language Teaching Research*, vol. 16, no. 1, pp. 89–108, 2012.

[2] D. J. Young, "Linguistic simplification of SL reading materials: Effective instructional practice?" *Modern Language Journal*, vol. 83, no. 3, pp. pp. 350–366, 1999.

[3] R. Chandrasekar and B. Srinivas, "Automatic induction of rules for text simplification," *Knowledge-Based Systems*, vol. 10, no. 3, pp. 183 – 190, 1997.

[4] S. Devlin and J. Tait, "The use of a psycholinguistic database in the simplification of text for aphasic readers." *Linguistic Databases*, pp. 161–173, 1998.

[5] S. Jonnalagadda and G. Gonzalez, "BioSimplify: an open source sentence simplification engine to improve recall in automatic biomedical information extraction," in *Annual Proceedings of AMIA 2010*, November 2010, pp. 13–17.

[6] B. B. Klebanov, K. Knight, and D. Marcu, "Text simplification for information-seeking applications," in *On the Move to Meaningful Internet Systems, Lecture Notes in Computer Science*. Springer Verlag, 2004, pp. 735–747.

[7] D. Vickrey and D. Koller, "Applying sentence simplification to the conll-2008 shared task," in *Proceedings of the Twelfth Conference on Computational Natural Language Learning*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 268–272.

[8] L. Feng, "Text simplication: A survey," CUNY, Tech. Rep., March 2008.

[9] M. Yatskar, B. Pang, C. Danescu-Niculescu-Mizil, and L. Lee, "For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 365–368.

[10] W. Coster and D. Kauchak, "Learning to simplify sentences using Wikipedia," in *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. Portland, Oregon: Association for Computational Linguistics, June 2011, pp. 1–9.

[11] C. Napoles and M. Dredze, "Learning Simple Wikipedia: A cogitation in ascertaining abecedarian language," in *NAACL HLT 2010 Workshop on Computational Linguistics and Writing*, 2010, pp. 42–50.

[12] O. Biran, S. Brody, and N. Elhadad, "Putting it simply: a context-aware approach to lexical simplification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 496–501.

[13] J. De Belder, K. Deschacht, and M.-F. Moens, "Lexical simplification," in *1st International Conference on Interdisciplinary Research on Technology, Education and Communication*, 2010.

[14] J. De Belder and M. Moens, "Text simplification for children," in *Prroceedings of the SIGIR workshop on accessible search systems*, 2010, pp. 19–26.

[15] S. Blum and E. A. Levenston, "Universals of lexical simplification," *Language Learning*, vol. 28, no. 2, pp. 399–415, 1978.

[16] J. E. Hoard, R. Wojcik, and K. Holzhauser, "An automated grammar and style checker for writers of simplified English," in *Computers and Writing*. Springer Netherlands, 1992, pp. 278–296.

[17] G. Adriaens, "Simplified English grammar and style correction in an MT framework: the LRE SECC project," in *Aslib proceedings*, vol. 47. MCB UP Ltd, 1995, pp. 73–82.

[18] A. Siddharthan, "Syntactic simplification and text cohesion," *Research on Language & Computation*, vol. 4, pp. 77–109, 2006.

[19] Z. Zhu, D. Bernhard, and I. Gurevych, "A monolingual tree-based translation model for sentence simplification," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010, pp. 1353–1361.

[20] S. Wubben, A. van den Bosch, and E. Krahmer, "Sentence simplification by monolingual machine translation," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 1015–1024.

[21] K. Woodsend and M. Lapata, "Learning to simplify sentences with quasi-synchronous grammar and integer programming," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 409–420.

[22] S. E. Petersen and M. Ostendorf, "Text simplification for language learners: A corpus analysis," in *Speech and Language Technology for Education workshop*, 2007.

[23] S. Kandula, D. Curtis, and Q. Zeng-Treitler, "A semantic and syntactic text simplification tool for health content," in *AMIA Annual Symposium Proceedings*. American Medical Informatics Association, 2010, pp. 366–370.

[24] H. Jing, "Sentence reduction for automatic text summarization," in *Proceedings of the sixth conference on Applied natural language processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 310–315.

[25] C. Blake, J. Kampov, A. K. Orphanides, D. West, and C. Lown, "Query expansion, lexical simplification and sentence selection strategies for multi-document summarization," in *Document understanding conference (DUC-2007)*, 2007.

[26] S. Silveira and A. Branco, "Combining a double clustering approach with sentence simplification to produce highly informative multi-document summaries," in *Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference*, Aug 2012, pp. 482–489.

[27] J. M. Conroy, J. G. Stewart, and J. D. Schlesinger, "Classy query-based multi-document summarization," in *Proceedings of the Document Understanding Conf. Wksp. 2005 (DUC 2005) at the Human Language Technology Conf./Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.

[28] A. Siddharthan, A. Nenkova, and K. Mckeown, "Syntactic Simplification for Improving Content Selection in Multi-Document Summarization," in *In Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004*, 2004, pp. 896–902.

[29] J. Kincaid, R. Fishburne, R. Rogers, and B. Chissom, "Derivation of new readability formulas (automated readability index, fog count, and flesch reading ease formula) for navy enlisted personnel," *Research Branch report*, 1975.

[30] G. H. McLaughlin, "Smog grading: A new readability formula," *Journal of reading*, vol. 12, no. 8, pp. 639–646, 1969.

[31] C. Napoles, B. Van Durme, and C. Callison-Burch, "Evaluating sentence compression: Pitfalls and suggested remedies," in *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. Portland, Oregon: Association for Computational Linguistics, June 2011, pp. 91–97.

[32] C. Scarton, M. de Oliveira, A. Candido, Jr., C. Gasperin, and S. M. Aluísio, "Simplifica: a tool for authoring simplified texts in brazilian portuguese guided by readability assessments," in *Proceedings of the NAACL HLT 2010 Demonstration Session*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 41–44.

[33] R. Barzilay and N. Elhadad, "Sentence alignment for monolingual comparable corpora," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 25–32.

[34] S. Bott and H. Saggion, "An unsupervised alignment algorithm for text simplification corpus construction," in *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. Portland, Oregon: Association for Computational Linguistics, June 2011, pp. 20–26.

[35] L. Specia, "Translating from complex to simplified sentences," in *Proceedings of the 9th International Conference on Computational Processing of the Portuguese Language*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 30–39.

[36] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proceedings of the second international conference on Human Language Technology Research*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 138–145.

[37] K. Inui, A. Fujita, T. Takahashi, R. Iida, and T. Iwakura, "Text simplification for reading assistance: A project note," in *Proceedings of the Second International Workshop on Paraphrasing*. Sapporo, Japan: Association for Computational Linguistics, July 2003, pp. 9–16.

[38] S. Devlin and G. Unthank, "Helping aphasic people process online information," in *Proceedings of the 8th international ACM SIGACCESS conference on computers and accessibility*. New York, NY, USA: ACM, 2006, pp. 225–226.

[39] N. Elhadad and K. Sutaria, "Mining a lexicon of technical terms and lay equivalents," in *Proceedings of the Workshop on BioNLP*

2007: Biological, Translational, and Clinical Language Processing. Association for Computational Linguistics, 2007, pp. 49–56.

[40] W. M. Watanabe, A. C. Junior, V. R. Uzêda, R. P. d. M. Fortes, T. A. S. Pardo, and S. M. Aluísio, "Facilita: reading assistance for low-literacy readers," in *Proceedings of the 27th ACM international conference on Design of communication*. New York, NY, USA: ACM, 2009, pp. 29–36.

[41] L. Deléger and P. Zweigenbaum, "Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora," in *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: from Parallel to Non-parallel Corpora*. Association for Computational Linguistics, 2009, pp. 2–10.

[42] S. Bott, L. Rello, B. Drndarević, and H. Saggion, "Can Spanish be simpler? LexSiS: Lexical simplification for Spanish," in *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, December 2012, pp. 357–374.

[43] B. Drndarević and H. Saggion, "Towards automatic lexical simplification in Spanish: An empirical study," in *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 8–16.

[44] H. Saggion, S. Bott, and L. Rello, "Comparing resources for Spanish lexical simplification," in *Statistical Language and Speech Processing*, ser. Lecture Notes in Computer Science, A.-H. Dediu, C. Martín-Vide, R. Mitkov, and B. Truthe, Eds. Springer, Berlin Heidelberg, 2013, vol. 7978, pp. 236–247.

[45] L. Specia, S. K. Jauhar, and R. Mihalcea, "SemEval-2012 task 1: English lexical simplification," in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, 7-8 June 2012, pp. 347–355.

[46] S. R. Thomas and S. Anderson, "WordNet-based lexical simplification of a document," in *Proceedings of KONVENS 2012*, J. Jancsary, Ed. ÖGAI, September 2012, pp. 80–88.

[47] R. Keskisärkkä, "Automatic text simplification via synonym replacement," Ph.D. dissertation, Linköping, 2012.

[48] G. Leroy, J. E. Endicott, D. Kauchak, O. Mouradi, and M. Just, "User evaluation of the effects of a text simplification algorithm using term familiarity on perception, understanding, learning, and information retention," *Journal of Medical Internet Research*, vol. 15, no. 7, p. e144, 2013.

[49] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[50] H. Kučera and W. N. Francis, *Computational analysis of present-day American English*. Providence, RI: Brown University Press, 1967.

[51] S. M. Aluísio and C. Gasperin, "Fostering digital inclusion and accessibility: the PorSimples project for simplification of Portuguese texts," in *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 46–53.

[52] S. Bautista, R. Hervás, P. Gervás, R. Power, and S. Williams, "A system for the simplification of numerical expressions at different levels of understandability." in *Worshop Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*, Atlanta, USA, 06/2013 2013.

[53] L. Rello, S. Bautista, R. Baeza-Yates, P. Gervás, R. Hervás, and H. Saggion, "One half or 50%? an eye-tracking study of number representation readability," in *Human-Computer Interaction INTERACT 2013*, ser. Lecture Notes in Computer Science, P. Kotzé, G. Marsden, G. Lindgaard, J. Wesson, and M. Winckler, Eds. Springer Berlin Heidelberg, 2013, vol. 8120, pp. 229–245.

[54] M. Brysbaert and B. New, "Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English," *Behavior Research Methods*, vol. 41, no. 4, pp. 977–990, 2009.

[55] T. Brants and A. Franz, "Web 1T 5-gram corpus version 1.1," *Linguistic Data Consortium*, 2006.

[56] K. Deschacht and M. Moens, "The latent words language model," in *Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning*, 2009.

[57] L. Rello, R. Baeza-Yates, L. Dempere-Marco, and H. Saggion, "Frequent words improve readability and short words improve understandability for people with dyslexia," in *Human-Computer Interaction INTERACT 2013*, ser. Lecture Notes in Computer Science, P. Kotz, G. Marsden, G. Lindgaard, J. Wesson, and M. Winckler, Eds. Springer Berlin Heidelberg, 2013, vol. 8120, pp. 203–219.

[58] D. Kauchak, "Improving text simplification language modeling using unsimplified text data," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 1537–1546.

[59] A. Ligozat, C. Grouin, A. Garcia-Fernandez, and D. Bernhard, "Approches à base de fréquences pour la simplification lexicale," in *Actes de TALN'2013 : 20e conférence sur le Traitement Automatique des Langues Naturelles*, vol. 1, Les Sables d'Olonne, France, 2013, pp. 493–506.

[60] C. Quirk, C. Brockett, and W. Dolan, "Monolingual machine translation for paraphrase generation," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 142–149.

[61] F. Rinaldi, J. Dowdall, K. Kaljurand, M. Hess, and D. Mollá, "Exploiting paraphrases in a question answering system," in *Proceedings of the second international workshop on Paraphrasing - Volume 16*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 25–32.

[62] D. Kauchak and R. Barzilay, "Paraphrasing for automatic evaluation," in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 455–462.

[63] A. Ligozat, C. Grouin, A. Garcia-Fernandez, and D. Bernhard, "Annlor: A naïve notation-system for lexical outputs ranking," in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, 7-8 June 2012, pp. 487–492.

[64] R. Sinha, "Unt-simprank: Systems for lexical simplification ranking," in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, 7-8 June 2012, pp. 493–496.

[65] S. K. Jauhar and L. Specia, "UOW-SHEF: SimpLex – lexical simplicity ranking based on contextual and psycholinguistic features," in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, 7-8 June 2012, pp. 477–481.

[66] M. Amoia and M. Romanelli, "Sb: mmsystem - using decompositional semantics for lexical simplification," in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, 7-8 June 2012, pp. 482–486.

[67] A. Johannsen, H. Martínez, S. Klerke, and A. Søgaard, "Emnlp@cph: Is frequency all there is to simplicity?" in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, 7-8 June 2012, pp. 408–412.

[68] J. Carroll, G. Minnen, Y. Canning, S. Devlin, and J. Tait, "Practical simplification of English newspaper text to assist aphasic readers," in *Proceedings of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, 1998, pp. 7–10.

[69] W. Daelemans, A. Höthker, and E. Sang, "Automatic sentence simplification for subtitling in Dutch and English," in *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004, pp. 1045–1048.

[70] S. Jonnalagadda and G. Gonzalez, "Sentence simplification aids protein-

protein interaction extraction," in *The 3rd International Symposium on Languages in Biology and Medicine*, November 2009, pp. 8–10.

[71] V. Seretan, "Acquisition of syntactic simplification rules for french," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012.

[72] B. T. Hung, N. L. Minh, and A. Shimazu, "Sentence splitting for Vietnamese-English machine translation," in *Knowledge and Systems Engineering (KSE), 2012 Fourth International Conference*, August 2012, pp. 156 –160.

[73] M. J. Aranzabe, A. D. de Ilarraza, and I. Gonzalez-Dios, "Transforming complex sentences using dependency trees for automatic text simplification in Basque," *Procesamiento del Lenguaje Natural*, vol. 50, pp. 61–68, 2012.

[74] G. Barlacchi and S. Tonelli, "Ernesta: A sentence simplification tool for childrens stories in Italian," in *Computational Linguistics and Intelligent Text Processing*, ser. Lecture Notes in Computer Science, A. Gelbukh, Ed. Springer Berlin Heidelberg, 2013, vol. 7817, pp. 476–487.

[75] J.-W. Chung, H.-J. Min, J. Kim, and J. C. Park, "Enhancing readability of web documents by text augmentation for deaf people," in *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*. New York, NY, USA: ACM, 2013, pp. 30:1–30:10.

[76] D. Feblowitz and D. Kauchak, "Sentence simplification as tree transduction," in *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 1–10.

[77] S. Klerke and A. Søgaard, "Simple, readable sub-sentences," in *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 142–149.

[78] S. Štajner, B. Drndarević, and H. Saggion, "Corpus-based sentence deletion and split decisions for Spanish text simplification," *Revista Computación y Sistemas; Vol. 17 No. 2*, 2013.

[79] B. Drndarević, S. Štajner, S. Bott, S. Bautista, and H. Saggion, "Automatic text simplification in spanish: A comparative evaluation of complementing modules," in *Computational Linguistics and Intelligent Text Processing*, ser. Lecture Notes in Computer Science, A. Gelbukh, Ed. Springer Berlin Heidelberg, 2013, vol. 7817, pp. 488–500.

[80] C. Gasperin, L. Specia, T. Pereira, and S. M. Aluísio, "Learning when to simplify sentences for natural text simplification." in *Encontro Nacional de Inteligência Artificial*, 2009, pp. 809–818.

[81] J. Jan, S. Damay, G. Jaime, D. Lojico, D. B. Tarantan, and E. C. Ong, "Simtext text simplification of medical literature," in *3rd National Natural Language Processing Symposium - Building Language Tools and Resources*, 2006, pp. 34–38.

[82] S. Eom, M. Dickinson, and R. Sachs, "Sense-specific lexical information for reading assistance," in *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 316–325.

[83] L. Rello, R. Baeza-Yates, S. Bott, and H. Saggion, "Simplify or help?: text simplification strategies for people with dyslexia," in *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. New York, NY, USA: ACM, 2013, pp. 15:1–15:10.

[84] W. M. Watanabe, A. Candido, Jr., M. A. Amâncio, M. de Oliveira, T. A. S. Pardo, R. P. M. Fortes, and S. M. Aluísio, "Adapting web content for low-literacy readers by using lexical elaboration and named entities labeling," in *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*. New York, NY, USA: ACM, 2010, pp. 8:1–8:9.

[85] H.-B. Chen, H.-H. Huang, H.-H. Chen, and C.-T. Tan, "A simplification-translation-restoration framework for cross-domain SMT applications," in *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, December 2012, pp. 545–560.

[86] S. Stymne, J. Tiedemann, C. Hardmeier, and J. Nivre, "Statistical machine translation with readability constraints," in *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013); Linkoping Electronic Conference Proceedings*, 2013, pp. 375–386.

[87] D. Klaper, S. Ebling, and M. Volk, "Building a German/simple German parallel corpus for automatic text simplification," in *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 11–19.

[88] A. Lopez, "Statistical machine translation," *ACM Comput. Surv.*, vol. 40, no. 3, pp. 8:1–8:49, Aug. 2008.

[89] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 177–180.

[90] R. Nelken and S. Shieber, "Towards robust context-sensitive sentence alignment for monolingual corpora," in *Proceedings of EACL 2006, the 11th Conference of the European Chapter of the ACL*, Trento, Italy, April 2006, pp. 3–7.

[91] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318.

[92] F. Dell'Orletta, S. Montemagni, and G. Venturi, "Read-it: assessing readability of Italian texts with a view to text simplification," in *Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 73–83.

[93] S. Štajner, R. Evans, C. Orasan, and R. Mitkov, "What can readability measures really tell us about text complexity?" in *Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*, 2012, pp. 14–21.

[94] S. Klerke and A. Søgaard, "Dsim, a Danish parallel corpus for text simplification," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 4015–4018.

[95] J. De Belder and M. Moens, "A dataset for the evaluation of lexical simplification," in *Computational Linguistics and Intelligent Text Processing*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2012, vol. 7182, pp. 426–437.

[96] E. Dale and J. Chall, "A formula for predicting readability: Instructions," *Educational research bulletin*, pp. 37–54, 1948.

[97] A. Max, "Writing for language-impaired readers," *Computational Linguistics and Intelligent Text Processing*, pp. 567–570, 2006.