Context Aware Textual Entailment

Soha Arab-Khazaeli
Louisiana State University and Agricultural and Mechanical College

CONTEXT AWARE TEXTUAL ENTAILMENT

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Interdepartmental Program in
Engineering Science

by
Soha A. Khazaeli
B.S., Shahid Beheshti University, 2001
M.S., Isfahan University, 2004
December 2015

ACKNOWLEDGMENTS

I express my sincere appreciation to my advisor Dr. Gerald M. Knapp for his support, guidance and constructive criticism during the past four and half years of graduate study. I am also thankful to Dr. Jianhua Chen, Dr. Jerry Trahan, and Dr. David Constant for being on my committee.

I would like to thank my fellow graduate students Dr. Leili Javadpour, Dr. Mehdi Khazaeli, Shahrzad Shirzad, and Alireza Kheirkhahan for the help they gave and the knowledge they shared, and the entire semantic analysis research group at LSU.

My gratitude and appreciation goes to Mr. Scott St. Cyr, Mr. and Mrs. Valiollahi, and Mr. and Mrs. Kuehne for their generosity and unconditional love. Along the educational path and throughout my life they have always offered words of encouragement and advice. Their kindness will never be forgotten.

The most substantial contributions to my entire education are, of course, from my parents, siblings Ali, Roja, Mehdi, my in-laws and friends who have been supportive and helpful throughout my life.

I thank my precious son, Faraz, for his sweet smile and his understanding. He has grown up watching me study and juggle with family and work. I am thankful to him for giving me happiness during the hard days.

Finally and most importantly, my deepest appreciation and love goes to my best friend and husband, Reza. He has been by my side throughout this entire process, providing words of encouragement and giving me a shoulder to lean on when I most needed it.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In conversations, stories, news reporting, and other forms of natural language, understanding requires participants to make assumptions (hypothesis) based on background knowledge, a process called entailment. These assumptions may then be supported, contradicted, or refined as a conversation or story progresses and additional facts become known and context changes. It is often the case that we do not know an aspect of the story with certainty but rather believe it to be the case; i.e., what we know is associated with uncertainty or ambiguity.

In this research a method has been developed to identify different contexts of the input raw text along with specific features of the contexts such as time, location, and objects. The method includes a two-phase SVM classifier along with a voting mechanism in the second phase to identify the contexts. Rule-based algorithms were utilized to extract the context elements.

This research also develops a new context-aware text representation. This representation maintains semantic aspects of sentences, as well as textual contexts and context elements. The method can offer both graph representation and First-Order-Logic representation of the text.

This research also extracts a First-Order Logic (FOL) and XML representation of a text or series of texts. The method includes entailment using background knowledge from sources (VerbOcean and WordNet), with resolution of conflicts between extracted clauses, and handling the role of context in resolving uncertain truth.

Keywords: context identification, context aware text representation, first-order-logic, textual entailment, and machine learning.

CHAPTER 1 – PROBLEM STATEMENT

Significant improvements in computational capabilities in recent years, coupled with an explosive growth in online text documents (news, e-books, social media, etc.) have led to growing interest in Natural Language Processing (NLP) research, and in particular Natural Language Understanding (NLU) which deals with the automated extraction of meaning from natural language text or speech.

A significant hurdle remaining for NLU is the area of textual entailment or natural language inference. Textual entailment is a directional relationship between a pair of text expressions: an entailing text (denoted T), and an entailed hypothesis (denoted H). T is said to entail H if the meaning of H can be inferred from the meaning of T, as would typically be interpreted by people (Dagan, Glickman, and Magnini 2006). To illustrate, Figure 1 shows an example of a Recognizing Textual Entailment (RTE) task. The "Text" entails Hyp1 and Hyp3, but not Hyp2.

Text: The Humane Society and SPCA are doing their best to rescue pets trapped in sweltering homes for more than a week. Getting them all will be impossible.

Hyp1: Some pets remained trapped in homes.

Hyp2: ASPCA tried to rescue trapped pets.

Hyp3: SPCA tried to rescue animals

Figure 1: RTE example

Many NLP problems can benefit from or require RTE capabilities. For example, in document summarization, textual entailment can be used to recognize and remove redundant sentences by checking whether a sentence can or cannot be entailed directly from previous text (Dang and Owczarzak 2008). Information Extraction (IE), the task of extracting instances of certain sets of templates in a text like “X born in Y” or “X capital of Y” can be converted to an entailment problem (Roth, Sammons, and Vydiswaran 2009). Question Answering (QA), which seeks to provide the best answer to a question based on a large collection of documents can use entailment to find the best answers among candidate answers (Harabagiu and Hickl 2006). New research in Machine Translation (MT) evaluation uses entailment to test whether the candidate translation entails (and is entailed by) the reference translation to estimate approximate semantic equivalence between candidate translation and reference translation (Padó et al. 2009). One of the most recently used applications of Textual Entailment is for Intelligent Tutor Systems (ITS), where entailment has been used to determine the extent to which a student’s answer to a question is entailed by the concept which is being taught (Nielsen, Ward, and Martin 2009).

Entailment can be applied in one of two directions: (1) starting with text and deducing hypothesis/hypotheses from the text (the focus of this research), or (2) starting with hypothesis/hypotheses and determining if it/they are entailed from a document or set of documents (the focus of RTE Challenge tasks).

Since 2005, an annual RTE Challenge has been run under PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning), a Network of Excellence (NoE) funded by the European Union. In 2008, the RTE Challenge became a track at the annual Text Analysis Conference (TAC). This challenge provides a uniform platform for comparing state-of-the-art

systems. Table 1 shows the performance of the five best participants in the RTE-7 main task held in 2011 (Bentivogli et al. 2011). The results show that task performance is still far from desired for practical application and indicate that more research is needed to achieve satisfactory performance. The joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge was held at the International Workshop on Semantic Evaluation in 2013. The task involved providing feedback on student answers based on reference answers. Although it required semantic inference and therefore is related to recognizing textual entailment (or paraphrasing), the system contained the extracted answer and there was no need to work on a complete text in this particular challenge.

Table 1: Five best participant performances in RTE-7 (main task)

Program	Precision (%)	Recall (%)	F-measure (%)	Reference
IKOMA	46.96	49.08	48.00	Tsuchida and Ishikawa 2011
U_Tokyo	46.84	43.58	45.15	Yokote, Tanaka, and Ishizuka 2011
BUPTTeam	45.02	44.95	44.99	Tan et al. 2011
CELI	41.88	46.56	44.10	Kouylekov, Bosca, and Dini 2011
BIU	41.81	44.11	42.93	Stern et al. 2011

Many different approaches have been used in textual entailment research, but they all contain certain shared components. In general, the steps of an RTE system include **preprocessing**, **enrichment**, **alignment**, and **inference**. In the preprocessing phase, RTE systems use various off-the-shelf annotators and analyzers for sentence and word segmentation, part of speech (POS)

tagging, dependency parsing, syntactic parsing, named entity recognition (NER), coreference resolution, and semantic role labeling. In the enrichment phase, other resources are often used to provide additional semantic tagging, including WordNet (Fellbaum 1998), VerbNet (Schuler 2005), VerbOcean (Chklovski, Timothy, and Pantel 2004), and FrameNet (Baker, Fillmore, and Lowe 1998).

The alignment component attempts to align the H (hypothesis) to a portion of T (text). The idea is that only a small piece of the text is potentially relevant to a hypothesis. Thus, aligners extract the portion of text which appears relevant to a hypothesis, reducing the search space and simplifying the inference process. De Marneffe et al. used supervised learning by using human annotated alignment data to train their aligner (2007). MacCartney and his colleagues (2008) proposed the MANLI aligner which used phrase-based edit distance and syntactic and semantic features for supervised learning by Microsoft Research (MSR) corpus (Brockett 2007). Other researchers have used different versions of edit distance, and similarity metrics such as WordNet-based word similarity (Hickl and Bensley 2007; Mehdad 2009). Graph-based aligners have been very successful as they can capture implicit syntax and semantic aspects, in addition to lexical aspects (Zanzotto, Dell'Arciprete, and Moschitti 2011). The graphs, based on the dependency parse tree of the text and hypothesis, which may be enriched by other resources, are compared together. The most similar part of the text graph and the hypothesis graph, or the portion of text which can be transformed to the hypothesis graph in a series of defined operations, is extracted as the aligned text part (Stern et al. 2011).

The inference component determines whether the starting hypothesis is entailed by, not entailed by, or contradicted by the text. Machine Learning (ML) techniques are the primary method

utilized by most research systems for the inference component. Some researchers have explored First Order Logic (FOL) or other more semantic-oriented methods, but these typically have been "fragile" and cannot handle many situations; their precision may be slightly higher but recall is dramatically lower (Bos and Markert 2005; Bos2013). The successful RTE systems in the RTE-7 Challenges to date have all used machine learning methods, but differ in various features and the extent to which these features are semantic-oriented.

One group of RTE systems research has focused on selecting the best features from (1) **lexical similarity**: Lin Similarity (Lin 1998), directional similarity (Kotlerman et al. 2010), similarity based on WordNet, VerbOcean, geographic resource, and n-grams (Heilman and Madnani 2013); (2) **syntactic similarity** : similarity on matched POS tag words only, similarity on chunk level (Tsuchida and Ishikawa 2011); and (3) **semantic similarity**: Discovery of Inference Rule from Text (DIRT) similarity (Lin and Pantel 2001), predicate-argument-structure features (Tsuchida and Ishikawa 2011;Tan et al. 2011; Kouylekov, Bosca, and Dini 2011), Named Entity similarity and time expression similarity (Rudzewitz and Ziai 2015).

A second body of research has focused on graph representation of the hypothesis and the text, both containing syntactic and semantic aspects of the reference document. Researchers have defined graph operations, as opposed to inference rules, which can be applied to the graphs. They also have defined various graph features (subgraph similarities, graph operations) to use in ML methods (Stern et al. 2011; Zanzotto, Dell'Arciprete, and Moschitti 2011).

A third group has explored different logic methods for text-inference, such as first-order logic (FOL), natural logic, and episodic logic. Bos and Markert (2005) used the following deep

semantic features extracted from the *theorem prover*[1] Vampire (Riazanov and Voronkov 2002) and from Paradox, a *model builder*[2] (Claessen and Sorensson 2003): entailed, inconsistent, domainsize, modelsize, domainsizeabsdif, domainsizereldif, modelsizeabsdif, modelsizereldif. Both programs operate based on first-order logic. MacCartney designed and implemented a natural language inference system (NatLog), based on natural logic theory (Lakoff 1972;van Benthem 1988). In NatLog, machine learning methods were used in some phases of inference (MacCartney 2009). EPILOG, an episodic logic system, uses semantic representation, which supports interpretive and inferential needs of natural logic understanding (Schubert and Hwang 2000). It uses probabilistic methods in some phases of inference, researchers also plan to apply ML methods to increase the scope and robustness of EPILOG (Schubert, Van Durme, and Bazrafshan 2010).

Context has a long history in different areas of artificial intelligence. Context defines state of the environment or world, and clauses may only be true in certain contexts. For instance, the assertion “Joe is hungry” may be true before a dinner event but false immediately after. The first formalization of context was offered by McCarthy (1993). Buvač and Mason (1994) proposed Propositional Logic of Context (PLC). Ghidini and Giunchiglia (2001) proposed MultiContext Systems (MCS) and argued that contextual reasoning can be analyzed as the result of the interaction of the principle of locality[3], and the principle of compatibility[4]. Dey (2001) identified four categories of context: identity, location, status and time. In spite of the long history of context

[1] A system able to prove theorems in first-order logic.

[2] A system provides finite-domain models for first-order problems.

[3] Reasoning always happens in a context.

[4] There can be relationships between reasoning processes in different contexts.

in other research areas and its importance in natural language understanding, it has not been considered in advanced text entailment systems.

Despite years of research in textual entailment, state-of-the-art techniques only slightly outperform pure statistical methods. The textual entailment methods suffer from a lack of deep semantics. In addition, most of them require alignment to decrease the problem complexity inherent in textual entailment challenges. Alignment causes local entailment which is in direct contrast to the goal of natural language understanding. In fact, alignment excludes any understanding that requires evaluating relations across more than one sentence. This research focuses on semantic and first-order-logic to obtain inference based on an overall understanding of the text.

1.1 Objectives

The major objectives of this study are:

- Develop a system to model document level entailment from story-style documents (news, novels) incorporating the following research advancements:
    - Development of a method for detecting context changes in documents.
    - Development of a FOL and XML representation model for expressing state, identity, temporal, and spatial knowledge from a document, including knowledge from background (common-sense) knowledge sources.

    - Development of a set of rules to enable entailment based on the FOPL representation using common reasoning rules and incorporating background knowledge and context within the rules.

- Performance analysis and method refinement of the methods to balance accuracy and semantic depth versus model complexity.

CHAPTER 2- LITERATURE REVIEW

In this chapter, the following text inference methodologies are reviewed: statistical methods, transformation based methods, and logic based methods. Context, definitions and relevant research are discussed followed by an overview of existing text representations. Additionally, two common sense knowledge resources are introduced.

2.1 Inference Based on Text

A critical task in textual entailment is inference. Various research teams utilize different methods to perform inference. These methods are categorized as statistical methods, transformation based methods, and logic based methods.

2.1.1 Statistical methods

Statistical methods and machine learning have been extensively utilized in textual entailment research. Generally, lexical, syntactical and semantic features are extracted for use in a machine learning method, though the selected features vary significantly from research to research. At TAC 2011, Tsuchida and Ishikawa (IKOMA) utilized lexical level, chunk level and predicate-argument-structure level information in a LibSVM classifier (2011). Yokote and his colleagues (2011), as team U-Tokyo, used WordNet Similarity::lch (Leacock and Chodorow 1998) and WordNet Similarity::jcn (Jiang and Conrath 1997), as well as term weights as features. Tan et al. (BUPTTeam) (2011) used the well-known *term frequency-inverse document frequency* (tf-idf) factor to feed their SVM-based classifier. Hickl and Bensley used Levenstein (2007) string-edit distance and named entity similarity along with other features. Zanzotto and colleagues (2011), using their SemKer system, used the similarity between pairs of text trees and pairs of hypothesis

trees extracted from entailment cases. Bär and his colleages added feature by Greedy String Tiling (GST)[5] along other feature in their system Ukp (2012). Bjerva and his colleagues (2014) used more semantic-oriented features such as agent-pair similarity and patient-pair similarity in Discourse Representation Structure (DRS). Rudzewitz and Ziai (2015) added named entity similarity and time expression similarity to their CoMiC system. Table 2 summarizes lexical, syntactical and semantic features which have been used in textual entailment systems.

Table 2: Text features used in textual entailment research

Feature	Description	Used By
Entailment score	A score based on WordNet[6]	Tuschida and Ishikawa 2011
Entailment score between words with same POS		Tuschida and Ishikawa 2011
Cosine similarity	Between text vector and hypothesis vector	Tuschida and Ishikawa 2011
Wikipedia	Log (m), where m is the estimated accuracy of the method used to learn the given Wikipedia rule, as described in (Shnarch, Barak, and Dagan 2009). $0 \leq m \leq 1$.	Stern et al. 2011
Lin similarity	Log (sim), where sim is the similarity score according to (Lin 1998). $0 \leq sim \leq 1$.	Stern et al. 2011
Lch similarity	WordNet Similarity (Leacock and Chodorow 1998)	Yokote, Tanaka, and Ishizuka 2011

[5] GST algorithm recognizes the longest sequence of substrings in the source document which is suspected to be plagiarized in another document. The algorithm returns the sequence as tiles from the source document and the suspicious document.

[6] Ent_sc(T, H)= $\frac{\sum_{t_h \epsilon H_t} match(t_h, T_{ht}, R) w(t_h)}{\sum_{t_h \in H_t} w(t_h)}$ $\qquad w(t) = (log\,\frac{N}{freq(t)})^{\alpha}$

Here, each T_t and H_t denote a set of words in each given text T and H. w(t) is the weight of the word t, and freq(t) is the weight of the word t in a corpus. N is the number of the texts in the corpus. R is a set of knowledge resources. Match(t,T_t, R) takes 1 if the word t corresponds to a word in T_t, otherwise takes 0.

Table 2 continued

Feature	**Description**	**Used By**
Jcn similarity	WordNet Similarity (Jiang and Conrath 1997)	Yokote, Tanaka, and Ishizuka 2011
WordNet-based similarity	(Pedersen, Patwardhan, and Michelizzi 2004)	Hickl and Bensley, 2007 , Rudzewitz and Ziai 2015
tf-idf factor		Tan et al. 2011
Levenstein string edit-distance	Edit-distance calculated by Levenstein algorithm	Hickl and Bensley, 2007, Rudzewitz and Ziai 2015
Directional similarity	Log (sim), where sim is the similarity score according to (Kotlerman et al. 2010) $0 \leq sim \leq 1$.	Tan et al. 2011
Named entity similarity		Hickl and Bensley 2007
DIRT	Log (sim), where sim is the similarity score according to (Lin and Pantel 2001). $0 \leq sim \leq 1$.	Stern et al. 2011
Graph similarity	Similarity between pairs of text trees and pairs of hypothesis trees	Zanzotto, Dell'Arciprete, and Moschitti 2011
Matching ratio for each chunk type	(e.g., NP and VP)	Tuschida and Ishikawa 2011
Matching ratio for each argument type		Tuschida and Ishikawa 2011
The number of negation mismatch in all corresponding Predicate Argument Structure (PAS)		Tuschida and Ishikawa 2011
The number of modal verb mismatch in all corresponding PAS pairs		Tuschida and Ishikawa 2011
Greedy String Tiling		Bär et al. 2012
agent-pair similarity and patient-pair similarity		Bjerva, et al. 2014
Time Expression Similarity		Rudzewitz and Ziai 2015

2.1.2 Transformation based methods

Transformation based systems attempt to transform the text (or aligned parts of the text) into the hypothesis. The steps utilized usually employ edit distance methods or graph methods. Tree edits distance for textual inference was first used by Punyakanok and his colleagues (2004) in a question answering application. Later, several teams applied the idea in the RTE Challenge (Kouylekov and Magnini 2005). Using Edit Distance Textual Entailment Suite (EDITS) software, edit distance was used to transform the text into the hypothesis using edit distance operation at the string, token, and tree level (Mehdad 2009).

De Salvo Braz et al. (2005) described an RTE system which attempts to transform a text graph to a hypothesis graph using hand-coded rules. The rules capture alternative expressions of information at the lexical, phrasal, syntactic and predicate-argument levels. Bar-Haim et al. and Stern et al. (2007; 2011) classified their system, BIUTEE, as a hybrid system. In the first phase, BIUTEE transforms the tree representation of the text to intermediate trees. The goal was to convert text trees to hypothesis trees, but in most cases it was found not to be possible. Therefore, the transformation steps use a set of lexical and syntactical rules to produce the intermediate tree. Then, the entailment phase extracts features from the intermediate tree and the hypothesis tree and, using a statistical method, decides on the entailment. Lien (2014) classified their system, UIO-Lien, as a graph-based method. They use a semantic representation formalism called Minimal Recursion Semantics (MRS) to represent the text and the hypothesis. Alignment of key nodes in the hypothesis MRS to nodes in the text MRS is indicator of entailment in their system.

One of the most frequently used approaches in transformation based methods is to assign to each substitution in word, phrase or tree level in texts a certain and defined cost. The costs

determine which substitution should be chosen in each step. In the final step, the total cost determines if the transformation with total cost X is a possible transformation from the text to the hypothesis (i.e., the hypothesis is entailed by the text) or not.

2.1.3 Logic based methods

Bos and Markert (2005) investigated adding semantic features to textual entailment using first order logic. Their research used deep semantic features extracted from Vampire, a theorem prover (Riazanov and Voronkov 2002) and Paradox, an FOL model builder (Claessen and Sorensson 2003). Using a statistical method, they applied the semantic features, along with other lexical and syntactical features.

The Boeing Language Understanding Engine (BLUE) system by Clark and Harrison (2009) is based on a formal logical approach. The system converts the text to a logical representation using a bottom-up chart parser. The logical form is then used to infer entailment by WordNet and Discovery of Inference Rule from Text (DIRT) rules. The system uses a Bag of Words (BoW) module if the logical module fails to infer or refute an entailment.

EPILOG, an episodic logic system by Schubert and Hwang using semantic representation, is a reasoning system which has been under development since 1990. EPILOG is based on a natural logic-like representation and inference mechanism (2010) and is designed for forward and goal-directed inference, but there has been no application of this system to textual entailment.

MacCartney (2009) designed and implemented NatLog, a textual entailment system based on natural logic. NatLog implements a new model which extends the monotonicity calculus of van Benthem and Sánchez-Valencia to the integration of semantic exclusion and implicativity.

MacCartney described an expressive set of entailment relations and the algebra employed to determine their joins.

2.2 Context

Context awareness, as a core feature of computing systems, emerged and has been employed since the early 1990s. Since then, researchers and computer scientists have proposed many different definitions, systems, and prototypes - in various applications - to employ the concept of context (Perera et al. 2014). The term context has been defined by many researchers (Brown 1995; Franklin and Flaschbart 1998; Hull, Neaves, and Bedford-Roberts 1997). Dey (2001) provided the most cited definition of context: “Context is any information that can be used to characterize the situation of an entity”. Many context features and context models have been proposed in varied applications (Strang and Linnhoff-Popien 2004). Schilit and his colleagues (1994) used key-value pairs to model context. Several research systems have employed markup scheme models (Knappmeyer et al. 2010; Held, Buchholz, and Schill 2002; Capra, Emmerich, and Mascolo 2003). Bauer et al. and Henricksen et al. (2003; 2003) used the graphical models such as Unified Modeling Language (UML) and Object-Role Modeling (ORM). Strang et al. and Wang et al. (2003; 2004) developed an ontology based model.

The first logic-based formalization of context was offered by McCarthy (1993) and Giunchiglia (1993). According to McCarthy, contexts can partition knowledge into limited sets that include locally true axioms. It is always possible to extrapolate from a local context to a more general context. Buvač and Mason (1994) proposed Propositional Logic of Context (PLC). PLC was an attempt to formalize McCarthy's ideas on context. Conversely, according to Giunchiglia, context is a tool to localize reasoning to a subset of facts, and a more global context for each local

context is not guaranteed (Giunchiglia, Maltese, and Dutta 2012). Ghidini and Giunchiglia (2001) proposed MultiContext Systems (MCS) and argued that contextual reasoning can be analyzed as the result of the interaction of the principle of locality and the principle of compatibility.

The concept of context can be used in natural language processing (NLP) to provide a more complete and unencumbered image of the text in any time and location. According to Miller (1998), context as used in NLP research and applications can be defined by WordNet as follows: "Discourse that surrounds a language unit and helps to determine its interpretation."

NLP research found context features are valuable data. Therefore, the context features are extracted and applied. The NLP systems that use context features define a length-specific window (e.g., several adjacent words or sentences) and extract the features in the designated window (Gabrilovich and Markovitch 2009; Lin, Ng, and Kan 2014). In more focused research, context is categorized as either local or global. Local context is a short word sequence surrounding the target word whereas global context includes the whole text (Socher et al. 2013; Huang et al. 2012). Local context is valuable, though not sufficient. Dethlefs and Cuayahuitl (2015) cited this particular issue in their research on natural language generation. Processing the entire text is time consuming. Moreover, some of the extracted features can be misleading because they don't belong to the same scene or semantic, and may even be in contradiction to the current point of the text. Context, as based on the above definition, is the largest semantic unit in the text that possesses a coherent and non-contradictory image of the text world, and usually it has a limited time period and a specific location. If this semantic unit (context) can be identified, its features can be very valuable in different applications of NLP.

In linear text segmentation, early approaches utilized linguistic information such as cue phrases, syntax or lexical features (Beeferman, Berger, and Lafferty 1999). The dominant direction in current text segmentation research is based on the idea that when the topic changes, the vocabulary also changes (Choi 2000; Malioutov and Barzilay 2006). Another recent notable research trend is using topic models for text segmentation (Misra et al. 2009; Du, Buntine, and Johnson 2013).

About 15 years ago, researchers began exploring scene detection in video segmentation. A scene is a section of a motion picture with unified time and space (Katz 1998), and is very close to the concept of context in this research. The goal is to find a semantically meaningful scene automatically (Del Fabro and Böszörmenyi 2013). The features available for the task include visual, audio and text. Zhai and her colleagues (2005) used text cues to construct a text feature along with other features to boost scene detection accuracy. Wang and his colleagues (2008) developed multimodal features including images, audio streams, and text transcripts to segment the videos, and reported the results according to the type of feature and the hybrid system used. The text features alone produced an F-measure of about 50 percent.

2.2.1 Identity

Identity describes the concept of assigning a unique identifier to an entity. An entity is any unique object or event. An entity may (and often does) exist across contexts.

2.2.2 Status

Status refers to the intrinsic characteristics of an entity within a context. Every status property assigned to an entity during the timeline of the text develops the status context. Generally these attributes are adjectives or adverbs.

2.2.3 Location

Context location includes places, geographical coordinates, elevation, and/or relative spatial relations (e.g., here, there, up, down, etc.). Named Entity Recognition (NER) systems extract locations within the text in addition to other entities (e.g., organization, person, etc.). Latitude and longitude of the location can be obtained via a geocoding API (e.g., Google) and added to the extracted information.

2.2.4 Time

Time is an important contextual feature and, as such, a standard representation of the expression and temporal order of events should be used. TimeML is a powerful specification language for event and temporal expressions in natural language text. It includes four major data structures: *Event*, *Timex3*, *Signal*, and *Link*. *Event* refers to the situation or event that occurred, including attributes such as the id, tense, aspect, etc. *Timex3* (an extension of Timex2) is used to markup explicit temporal expressions, including times, dates, durations. Figure 2 shows Timex3 tags for "John begins teaching one week from September 15".

```
John begins teaching
<TIMEX3 tid="t1" type="DURATION" value="P1W"
beginPoint="t2" endPoint="t3">
one week
</TIMEX3>
from
<TIMEX3 tid="t2" type="DATE" value="XXXX-9-15">
September 15
</TIMEX3>
<TIMEX3 tid="t3" type="DATE" value="XXXX-9-22"
temporalFunction="TRUE" anchorTimeID="t1" />
```

Figure 2: Timex3 representation (Pustejovsky et al. 2003)

.

Signal is a tag to show function words, which indicate temporal relation. In Figure 3, “every” is a signal. Link tags encode the different relations that exist between the temporal elements of a text. Link tags include TLink[7], Slink[8], and ALink[9]. Figure 3 is a TimeML representation of “John taught 20 minutes every Monday”.

State-of-the-art approaches to temporal challenges exhibit high accuracy in temporal expression extraction (Timex3) and event detection, but low accuracy in temporal relation detection. Table 3 summarizes the top accuracy performances reported for the TempEval3 Challenge in 2013 (UzZaman et al. 2012).

---

[7] A link representing temporal relations including before, after, includes, is-included, holds, simultaneous, after, before, identity, begins, ends, begun-by, and ended-by.

[8] A subordination link used for contexts introducing relations including modal, negative, evidential, neg-evidential, factive, and counter-factive.

[9] A link representing the relationship between an aspectual event and its argument event including initiates, culminates, terminates, and continues.

```
John
<EVENT eid="e1" class="OCCURRENCE"
     tense="PAST" aspect="NONE">
taught
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"
     signalID="s1" cardinality="EVERY"/>
<TIMEX3 tid="t1" type="DURATION"
     value="PT20M">
20 minutes
</TIMEX3>
<SIGNAL sid="s1">
every
</SIGNAL>
<TIMEX3 tid="t2" type="DATE"
     value="XXXX-WXX-1">
Monday
</TIMEX3>
<TLINK eventInstanceID="ei1"
     relatedToTime="t1" relType="HOLDS"/>
<TLINK eventInstanceID="ei1"
     relatedToTime="t2"
     relType="IS_INCLUDED"/>
```

Figure 3: TimeML representation (Pustejovsky et al. 2003)

Table 3: Performance of temporal tagging systems

Task Name	**Task**	**Approximate Accuracy of the Best Systems**
A	Temporal expression	90%
B	Event extraction	80%
C	Temporal relation (only)	55%
ABC	Temporal awareness (A, B & C)	30%

2.3 Text Representation

Many different approaches have been used in text representation research, but they can be categorized into two major groups. The first group is based on statistics, and the second group, comprised mainly of linguistics experts as opposed to text mining experts, is based on graphs.

Bag of Words (BoW) is a basic methodology of statistical text representation models. BoW is used to track any word in the text, along with its frequency. This method of text representation is called a Vector Space Model (VSM) (Salton, Wong, and Yang 1975). Another text representation model, N-grams, records the frequency of each n-word, rather than each word. In this particular usage, the VSM functions as a unigram model (Brown et al. 1992). To address the shortcomings of previous approaches, researchers assigned appropriate weights to the terms, a process known as term weighting. The most common method of term weighting is term frequency-inverse document frequency (tf-idf) (Lan et al. 2009). Latent Semantic Indexing (LSI) uses statistical methods to glean the most representative features of a document (Wei, Yang, and Lin 2008). Locality Preserving Indexing (LPI) discerns the characteristic features of the document (Cai et al. 2007). These statistical text representation methods work well for certain Natural Language Processing (NLP) problems such as text categorization.

Many computational linguistic experts, however, rely on more semantic-oriented text representation methods (Mihalcea and Radev 2011). The idea of using graphs as conceptual representations for language units and their relationships originated with early work in the field of psychology by Sigmund Freud and psycholinguists such as Schvaneveldt and Spitzer. Although the statistical representation of documents used in some NLP applications generates reasonable accuracy, a large amount of syntactical and semantic information is lost when sentences and

paragraphs are reduced to a vector. Advanced level NLP problems requiring Natural Language Understanding (NLU) (e.g., Text Entailment) can benefit from a more semantic-oriented text representation. For such applications, computational linguists turned to constituency-based and dependency-based parse trees.

2.3.1 Dependency graphs

Computational linguists developed tools of generating parse trees automatically both as a means of analyzing sentence structure and as representations of text (De Marneffe et al. 2007). Dependency-based parse graphs (parse trees based on dependency grammar) are used in various NLP research (Heilman and Smith 2010; Schmitz et al. 2012; Socher et al. 2014). They derive from the thematic relation of sentence part to verb and are usually represented by graphs. A major problem for early researchers was the complexity of the resulting structures, and hence any application was limited. This has changed in recent years, and now graph-based text representations are widely used. Figure 4 depicts a dependency graph for the sentence “Bell, based in Los Angeles, makes and distributes electronic, computer and building products”.

2.4 Common Sense Knowledge

Common sense, or background knowledge, plays an important role in entailment. Synonyms, paraphrasing, and common sense facts/relations are considered background knowledge. For example, if somebody reads “An earthquake happened in San Francisco in 1906”, they would likely be able to answer the question “Did an earthquake happen in California in 1906” because they have the unstated background knowledge: San Francisco is located in California. In this

research, WordNet (Fellbaum 1998) and VerbOcean (Chklovski and Pantel 2004) will be used as common sense knowledge sources.

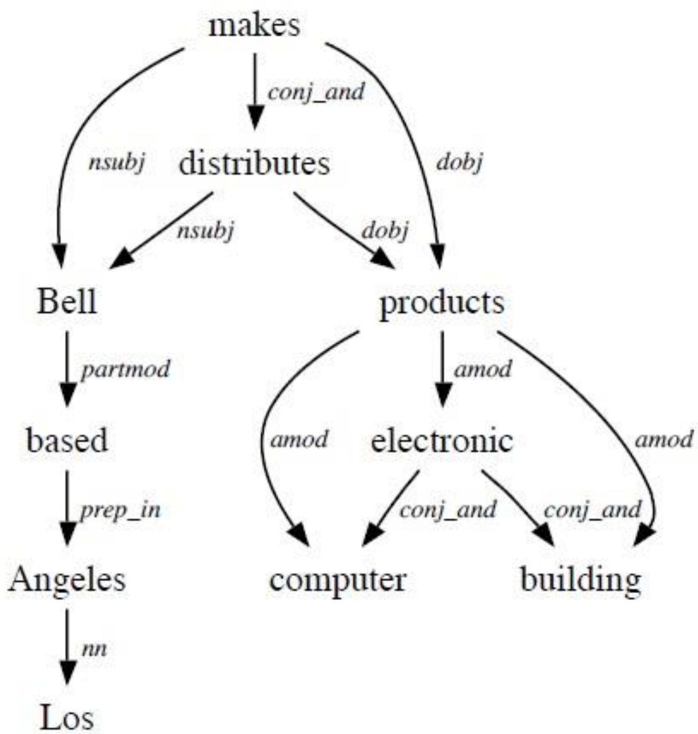

Figure 4: Dependency graph (De Marneffe and Manning 2008)

2.4.1 WordNet

WordNet is a large lexical database of the English language. Nouns, verbs, adjectives and adverbs are assembled into sets of cognitive synonyms (synsets), each expressing a distinct concept. Each word appears in one or more synsets, reflecting the ambiguity of words in natural languages. In some cases (e.g., antonyms), a relation exists only between two words, and not the entire synset. Synsets are interconnected by means of conceptual-semantic and lexical relations, and can be visualized as a graph with edges between words bearing some relationship. Such relationships

include hypernyms[10], hyponyms[11], coordinate terms[12], holonym[13], meronym[14], troponym[15], entailment[16], related nouns, similar to, participle of verb, and root adjectives.

2.4.2 VerbOcean

VerbOcean is a semantic network of verbs comprised of 3,477 verbs as nodes and 22,306 relations as edges. VerbOcean groups the relationship between two verbs into one of five categories. Table 4 depicts VerbOcean semantic relations.

Table 4: Semantic relations identified in VerbOcean (Chklovski and Pantel 2004)

Semantic Relation	**Example**	**Alignment with the WordNet**	**Symmetric**
Similarity	transform :: integrate	synonyms or siblings	Y
Strength	wound :: kill	synonyms or siblings	N
Antonymy	open :: close	antonymy	Y
Enablement	fight :: win	cause	N
Happens-before	buy :: sell marry :: divorce	cause, entailment, no temporal inclusion	N

[10] *Y* is a hypernym of *X* if every *X* is a *Y*.

[11] *Y* is a hyponym of *X* if every *Y* is an *X*.

[12] *Y* is a coordinate term of *X* if *X* and *Y* share a hypernym.

[13] *Y* is a holonym of *X* if *X* is a part of *Y*.

[14] *Y* is a meronym of *X* if *Y* is a part of *X*.

[15] The verb *Y* is a troponym of the verb *X* if to *Y* is to *X* in some manner.

[16] The verb *Y* is entailed by *X* if by doing *X* you must be doing *Y*.

VerbOcean pairs verbs according to their semantic relations. A numeric weight is assigned to each verb pair indicating the degree of the relation. Figure 5 depicts part of a VerbOcean file.

```
abandon [similar] reject :: 12.048992
abandon [similar] scrap :: 13.725957
embrace [happens-before] abandon :: 13.295483
abandon [opposite-of] embrace :: 10.723003
oppose [stronger-than] abandon :: 12.308010
abandon [opposite-of] adopt :: 10.410794
pursue [happens-before] abandon :: 11.262514
```

Figure 5: A part of VerbOcean file

Discovery of Inference Rules from Text (DIRT) is both an algorithm and a resulting knowledge collection. The algorithm automatically learns paraphrase relations (relations between synonymous expressions) between nouns using a dependency tree. If two different expressions tend to represent a binary relationship between the same set of nouns, DIRT concludes the two different expressions are paraphrase. The DIRT algorithm, mining a 1GB set of newspaper text (San Jose Mercury, Wall Street Journal, and AP Newswire from the TREC-9 collection) generated the following Top-20 paraphrases for “X solves Y”:

Y is solved by X, X resolves Y, X finds a solution to Y, X tries to solve Y, X deals with Y, Y is resolved by X, X addresses Y, X seeks a solution to Y, X does something about Y, X solution to Y, Y is resolved in X, Y is solved through X, X rectifies Y, X copes with Y, X overcomes Y, X eases Y, X tackles Y, X alleviates Y, X corrects Y, X is a solution to Y, X makes Y worse, X irons out Y.

In this research, VerbOcean is used. While DIRT shows relations generically, VerbOcean identifies relations explicitly and also provides frequency weights for each pair.

2.5 Related Corpora

The research described in this dissertation requires a corpus with features necessary for textual entailment. There are a number of textual entailment-related corpora such as FraCaS Corpus (Cooper et al. 1996) and RTE-7 Corpus (Dzikovska et al. 2013) but these corpora were developed for reverse entailment (entailment proving) rather than the forward entailment (entailment generation) addressed in this research. This research also requires a corpus with context tags but no such corpus exists. Hence a corpus is constructed for this research.

CHAPTER 3- APPROACH

This research develops a forward context aware textual entailment method. A schematic overview of the methodology is shown in Figure 6. The following sub problems were resolved in the development of the model:

- Development of a method for detecting context changes in documents.
- Development of a method for extracting contexts and their elements in documents.
- Development of the necessary preprocessing steps to generate a graph representation of the document supporting semantic hierarchy and the entailment system and incorporating background knowledge.
- Development of methods for converting the graph to first-order-logic statements and also xml structured files.
- Implementation, measurement and analysis of results, and refinement of the entailment system.

This research aims to provide a forward textual entailment method which is “aware” of context changes in text. Chapter 4 explains how the proposed system identifies changes for two classes of contexts: fine-grain and coarse-grain. A fine-grain context (simply called context) is a cohesive and consistent block of input text consisting of several sentences. Sentences in fine-grain context generally have strong semantic relations. Coarse-grain context is called scene in this research. A coarse-grain context or scene consists of several fine-grain contexts and usually takes place in a specific time and/or location.

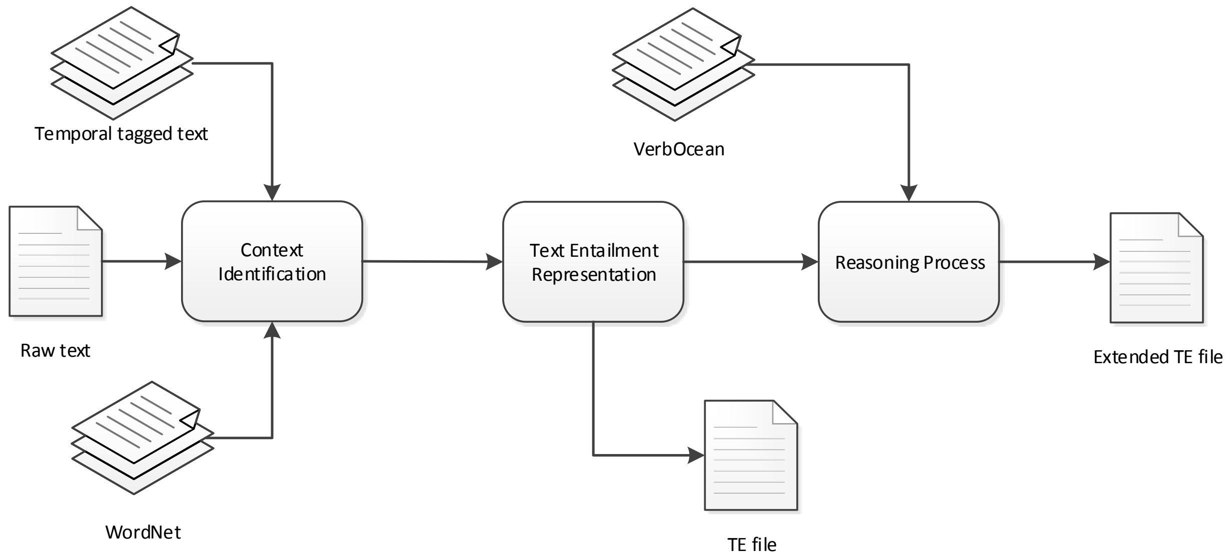

Figure 6: Context aware textual entailment overview

Chapter 5 describes a knowledge representation model that incorporates the hierarchical context semantics of a text while maintaining semantic detailed information of text elements. Chapter 6 expounds on reasoning and producing extra knowledge.

3.1 Tools

The following tools and programming language are used in the methodology and evaluation of this research: Java 1.8[17] in NetBeans IDE 8.0.2[18]; LibSVM 3.18 (Chang and Lin 2011); CoreNLP 3.5.0[19] (the Stanford NLP Package) (Manning et al. 2014) including word segmentation, POS tagging, named entity recognition, coreference resolution, time expression extraction; WordNet

[17] http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

[18] https://netbeans.org/

[19] http://nlp.stanford.edu/software/corenlp.shtml

Similarity Package[20] (Pedersen, Patwardhan, and Michelizzi 2004); Google GeoCode API[21]; TIPSemB[22] (Llorens, Saquete, and Navarro 2010) C99 UIMA Text Segmenter[23] (Choi 2000) Babelfy[24] (Moro, Cecconi, and Navigli 2014).

3.2 Corpus

To provide the basis for an absolute (rather than comparative) evaluation, a human annotated (gold-standard) corpus was created consisting of 20 English-language documents. Two reviewers annotated the documents. Inter-annotator agreement is calculated for context (scene) identification based on Cohen's kappa coefficient:

$$\kappa = \frac{\Pr(o) - \Pr(e)}{1 - \Pr(e)} \qquad \textbf{(Eq. 1)}$$

Where Pr(o) is the relative observed agreement among annotators, and Pr(e) is the hypothetical probability of chance agreement. The calculated kappa is equal with 68.82% which categorizes the agreement as a substantial agreement. There are 1716 sentences in the corpus. Each sentence can be the beginning of a scene or not. Two reviewers agreed on this decision on 1633 sentences, so Pr(o) is 1633/1716=95.16%. The hypothetical probability of chance agreement is 84.35%. Reviewer "A" categorized sentences as the end of scene 9.73% of the time, and Reviewer "B" categorized sentences as the end of a scene 7.34% of the time. Thus, the probability both of them

[20] https://wn-similarity.sourceforge.net/

[21] https://developers.google.com/maps/documentation/geocoding/

[22] http://timexportal.wikidot.com/tipsem

[23] https://code.google.com/p/uima-text-segmenter/

[24] https://babelfy.org/

would categorize a sentence as the end of a scene randomly is 9.73%*7.34%=0.71%, and the probability that both of them would not categorize a sentence as the end of a scene randomly is (1-9.73)% * (1-7.34)% = 83.64%. Thus the overall probability of random agreement is Pr(e) = 0.71% + 83.64% = 84.35%. Most of the disagreements were because one of the reviewers marked smaller and more specific scenes. Although Cohen's kappa coefficient is a standard metric for measuring inter-annotator agreement for NLP corpora, but it doesn't completely fit for text segmentation problems. Therefore, the P_k Error also is calculated between the two reviewers' corpus. The P_k Error between two corpora is calculated as 12.19%. Table 5 shows the result of two different ways of calculating inter-annotator agreement.

Table 5: Two different measures for inter-annotator agreement

Cohen's kappa coefficient (%)	P_k Error (%)
68.82	12.19

Conflicts between the two annotated set of texts were resolved by a third reviewer by voting among the three reviewers. GATE[25] Annotation (Cunningham et al. 2002) was used for annotation, communication, and review in the creation process. The dataset includes 135 scene contexts, 153 scene context changes, and 1,716 sentences. The number of scene context changes is increased 1 unit by appearance of the beginning of a different scene context. The number of scene changes are more than number of scenes because some of the scenes include several segments in different parts of the stories. Figure 7 shows a document in the corpus and Table 6 provides information about the stories in the corpus.

[25] https://gate.ac.uk/overview.html

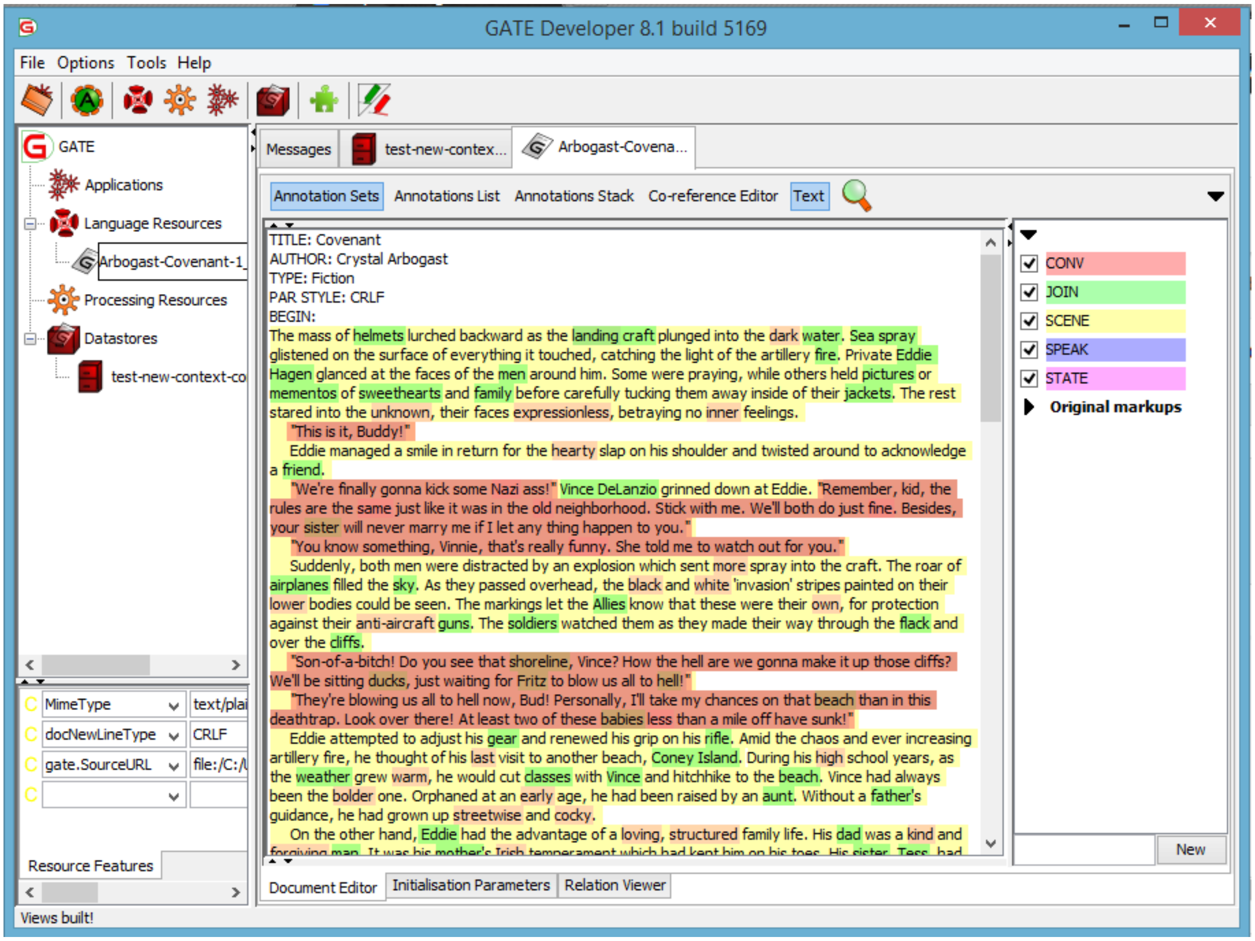

Figure 7: A tagged document in the corpus

Table 6: Information about the short stories in the corpus

Title	Author	Genre	Year	Contexts #	Sentences #	Words #
Covenant	Crystal Arbogast	Fiction	2000	6	138	1652
Hobnail	Crystal Arbogast	Fiction	2001	4	133	1669
The Whale Sound	Roger Dean Kiser	Fiction	-	6	43	983
The Three Fisherman	Tom Sheehan	Fiction	-	3	106	1950
Three Years	Katya AaltoTanssija	Non-Fiction	-	3	57	799
Shame	Amy Monticello	Non-Fiction	2014	3	41	769
Ten Years Ago	Sarah Beth Childers	Non-Fiction	2014	4	46	737
Like Momma	Lonette Stayton	Non-Fiction	2014	2	44	548
The Return	Fernando Sorrentino	Science Fiction	-	7	96	1919
The Star	Esther Claes	Science Fiction	-	6	63	1084
The Adventures Of Aladdin	The Brothers Grimm	Fairy Tale	-	12	146	1826
Beauty and The Beast	Unknown	Fairy Tale	-	13	117	1413
The Adventures of Thumb Tom	Unknown	Fairy Tale	-	7	62	835
The Country Mouse and The Town Mouse	Unknown	Fairy Tale	-	2	58	674
The Mighty Monster Afang	Unknown	Fairy Tale	-	6	108	2267
The King's Treasure	Unknown	Fairy Tale	-	13	118	2217
The Lady of The lake	Unknown	Fairy Tale	-	13	114	2208
The Touch of The Clay	Unknown	Fairy Tale	-	11	93	1797
The Fairy Congress	Unknown	Fairy Tale	-	2	78	1663
The Sword of Avalon	Unknown	Fairy Tale	-	7	55	674

CHAPTER 4- CONTEXT IDENTIFICATION

In this chapter, a two-phase technique is presented to identify scene-level contexts within a text. After identifying the contexts, the context elements (including time, location, objects, states, and dialogue) are extracted.

The raw text is first preprocessed to obtain POS tags, dependency graphs, coreferences, name entities, times, events, and their respective relations. Based on this information, 16 features are extracted. A trained supervised classifier will determine whether a particular sentence belongs to the previous context, or is the start of a new context. Fine-grain contexts (here after referred to simply as "contexts") are determined based on the features. A second trained supervised classifier with a cached voting mechanism will determine the coarse-grain contexts (here after referred to as "scenes").

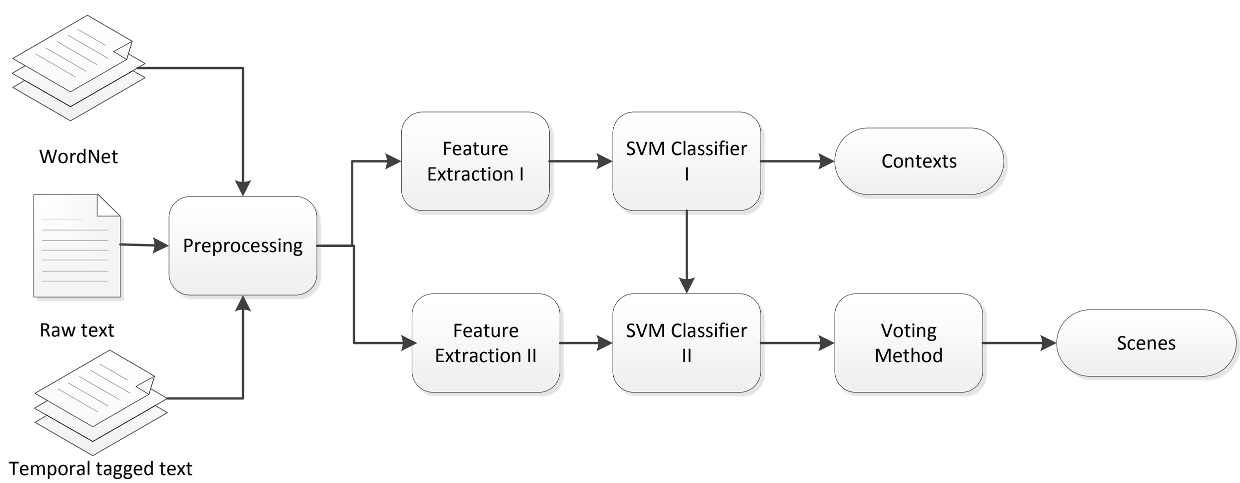

Figure 8: Context identification overview

4.1 Preprocessing

The raw text goes through preprocessing including sentence segmentation, word segmentation, POS tagging (Toutanova and Manning 2000; Toutanova et al. 2003) and sentence parsing (Klein and Manning 2003). These tasks are executed by CoreNLP, developed by the Stanford natural language processing group (Manning et al. 2014). Preprocessing output includes a dependency graph for each sentence (De Marneffe, MacCartney, and Manning 2006) a coreference graph (Lee et al. 2011; Lee et al. 2013) and named entities of input text (Finkel et al. 2005). The coreference graph represents a chain of mentions in textual order, showing all words that coreference to one word. The named entity classifier recognizes person, location, organization, miscellaneous, date, time, money, and number. A very important feature of context is *time*. TimeML standard rules are used for time expressions and the temporal order of events. Temporal tags are provided using the temporal information processing system TIPSemB (Llorens, Saquete, and Navarro 2010).

4.2 Classifying Fine-Grain Contexts (Contexts)

A classifier determines whether a particular sentence belongs to: (1) the previous context, or (2) the starting point of a new context.

4.2.1 Features

A group of features is extracted to classify each sentence as part of the previous context, or the start of a new context.

Time Similarity (F1) shows the similarity of the time in current context and previous context:

$$F1 = \begin{cases} 1 & if\ the\ two\ valid\ times\ are\ different \\ 0.5 & if\ one\ of\ the\ times\ are\ empty \\ 0 & otherwise \end{cases} \quad \textbf{(Eq. 2)}$$

The existence of a time expression in a sentence, as well as time differences between two sentences, can be an indication of a new context. For example, in "*During his <u>high school years</u>, as the weather grew warm, he would cut classes with Vince and hitchhike to the beach*", the time expression (underlined) signals the start of the new context.

Location Similarity (F2) shows the similarity of the location in current context and previous context:

$$F2 = \begin{cases} 1 & if\ the\ two\ valid\ locations\ are\ different \\ 0.5 & if\ one\ of\ the\ locations\ are\ empty \\ 0 & otherwise \end{cases} \quad \textbf{(Eq. 3)}$$

A location in a sentence can be an indication of a new context. For example, "*Uncle Jack had left the county, as well as the state of Virginia*".

Time Type (F3) sorts the time of current context by duration and time point:

$$F3 = \begin{cases} 1 & if\ the\ time\ is\ an\ explicit\ time\ or\ date \\ 0 & if\ the\ time\ is\ duration \end{cases} \quad \textbf{(Eq. 4)}$$

Research suggests the hypothesis that time points increase the probability of a new context more than duration.

Time Position (F4) shows the location of the time expression within a sentence:

$$F4 = \frac{\text{time expression position index}}{\text{number of words in the context}} \quad \textbf{(Eq. 5)}$$

Generally, if the time expression appears in the first words of the sentence, it is probably a new context.

Coreference Ratio (F5):

$$F5 = \frac{|coreferences\ between\ two\ contexts|}{|objects\ in\ current\ context|} \quad \textbf{(Eq. 6)}$$

A coreference between two parts of text indicates a strong connection between parts and decreases the probability of a new context.

Transition (F6):

$$F6 = \begin{cases} 1 & if\ \text{"before"}\ or\ \text{"after"}\ \text{temporal link exists between two contexts} \\ 0 & otherwise \end{cases} \quad \textbf{(Eq. 7)}$$

Temporal links such as *includes, hold*, and *simultaneous* indicate increased concurrency and coherency, and decrease the probability of a new context.

Special expression (F7):

Existence of the following expressions causes F7 to be 1, otherwise 0:

Once upon a time

hours\days\weeks\months\... later

After several hours\days\weeks\months\...

Later on

From this time forth

…

These expressions are indications of a new context.

Paragraph Start (F8):

$$F8 = \begin{cases} 1 & if\ it\ is\ beginning\ of\ a\ paragraph \\ 0 & otherwise \end{cases} \quad \textbf{(Eq. 8)}$$

Context switches often occur at the start of a new paragraph, although many times new paragraphs are not context switches.

Conversation Ratio (F9):

$$F9 = \frac{|conv_{sent}|}{|curCtx_{sent}|} \quad \textbf{(Eq. 9)}$$

where $|conv_{sent}|$ is the number of dialogue sentences in the current context and $|curCtx_{sent}|$ is the number of sentences in the current context. The conversations take place within the current context, and they usually do not change the context.

Previous Conversation Ratio (F10):

$$F10 = \frac{|preConv_{sent}|}{|prevCtx_{sent}|} \quad \textbf{(Eq. 10)}$$

where $|preConv_{sent}|$ is the number of dialogue sentences in the previous context and $|prevCtx_{sent}|$ is the number of sentences in the previous context.

Common Object (F11):

$$F11 = \begin{cases} 1 & if\ at\ least\ one\ object\ exists\ in\ both\ contexts \\ 0 & otherwise \end{cases} \quad \textbf{(Eq. 11)}$$

The term *object* means a tangible and physical entity. To identify such objects, a recursive algorithm is used to check the ancestors of the noun and its hypernym hierarchy. This algorithm is described in section 4.5.2. If two parts of the text have common objects, it is more likely that the two parts belong to one context.

Wordnet Similarities (F12-F16): Includes Wu-Palmer Similarity (Wu and Palmer 1994), Leacock-Chodorow Similarity (Leacock and Chodorow 1998), Jiang-Conrath Similarity (Jiang and Conrath 1997) Resnik Similarity (Resnik 1995) and Lin Similarity (Lin 1998). WordNet can only be used for words with identical POS tags. Thus the WordNet similarity is the average of all Wordnet similarities between matching pairs from two contexts:

$$sim_{ij} = \frac{\sum sim_{w_{i,k},w_{j,l}}}{n} \qquad \textbf{(Eq. 12)}$$

where $w_{i,k}$ is k^{th} word in i^{th} context, $sim_{w_{i,k},w_{j,l}}$ is WordNet similarity between two words, sim_{ij} is average WordNet similarity between two contexts, and n is the number of paired words.

4.2.2 Classifier

The feature vectors related to current and immediately previous sentences (context) are inputs to a classifier, which then determines whether two contexts should merge (class 0) or the current context is the start of a new context (class 1).

LibSVM (Chang and Lin 2011) proven to be powerful and efficient in NLP applications, has been chosen as the classifier to be trained in this research. The feature vectors include both positive and negative instances and were filtered to eliminate redundancy. Several negative instances were randomly removed to balance the number of positive and negative instances.

4.3 Classifying Coarse-Grain Contexts (Scenes)

In the first phase output, contexts recognized are coherent semantic units, but they are smaller than human-annotated contexts. Non-linear temporal events such as flash-forward and flashback often

produce blocks of sentences that are within the same context but are in non-adjacent positions within the text. To handle this, a classifier was developed for a second phase of categorization.

In phase two, the classifier determines whether or not any two recently specified contexts should merge. The feature vector elements for this phase are similar to the previous phase, except that in phase one, all features were calculated based on the current sentence and the previous sentence, whereas in this phase all features are calculated based on two input contexts. Each context now has a number of probable contexts with which it can merge. For this determination, a voting method considering local priority is used. This method merges any two adjacent contexts that the classifier has previously deemed merge-able, thus resolving the smaller than judged context problem. The adjacency problem caused by flash-forward and flashback is also resolved: any two contexts having greater intersection between their probable merge-able contexts and also are closer together rather than other candidates should be merged. The algorithm is shown in Figure 9.

4.4 Context Identification Evaluation

The context identifier in this research splits the text into segments. P_k metric is the standard measure used to assess text segmentation algorithms (Beeferman, Berger, and Lafferty 1999). P_k is an error metric indicating the percent error of the text segmentation. To calculate P_k, k is set to half the average of true segment size, and then penalties computed by a moving window of length k. At each location in the reference segmentation it is determined whether the start and end of the window occurs in the same segment. Same task is done on the classified segmentation concurrently. If the classified segmentation disagrees with reference segmentation, the counter will be increased. The P_k error is the average of the counter over the sliding window, and its value is

between 0 and 1. P_k error is calculated for the both phases of context identification. Table 7 shows an evaluation of the context identification process.

```
for each context in the contexts {
  if context in prevContext.PositiveCtx*
    Merge(preContext, context)
}
for each ctx1 in contexts
  for each ctx2 in contexts
    if (ctx1 <> ctx2) {
      intersectRatio = intersection(ctx1.PositiveCtx,ctx2.PositiveCtx).count /
                                  union(ctx1.PositiveCtx,ctx2.PositiveCtx).count
      if ctx2 has highest intersectRatio and closest to ctx1
        Merge(ctx1,ctx2)
    }
* PositiveCtx are the contextswhich are probable for merge by classifier.
```

Figure 9: Merging the contexts

Table 7: Context identification evaluation by P_k metric

Context Identification Phase	**K**	**P_k error**
I	6	0.30
II	6	0.28

The result obtained is then compared to a standard text segmentation algorithm known as C99 (Choi 2000) which is implemented in UIMA[26] (Unstructured Information Management Architecture). The comparison is depicted in Table 8.

[26] https://uima.apache.org/external-resources.html

Table 8: P_k error comparison

Context Identification Phase	P_k error
C99	0.42
I	0.30
II	0.28

In phase I, coherent semantic segments are developed, but the size of the contexts produced is smaller than the judged context. To evaluate this claim, the classified contexts are investigated using Equation 13.

$$C_{sim} = \frac{\sum_{t=1}^{m} \sum_{i=1}^{n} \frac{\left|C_{classified_{i,t}} \cap C_{judged_{j,t}}\right|}{\left|C_{classified_{i,t}}\right|}}{m * n} \qquad \textbf{(Eq. 13)}$$

where $C_{classified_{i,t}}$ is the i[th] context classified in t[th] text with the classifier, $C_{judged_{j,t}}$ is the most similar judged context to $C_{classified_{i,t}}$, n is the number of classified contexts, m is the number of texts, and $\left|C_{classified_{i,t}} \cap C_{judged_{j,t}}\right|$ is the number of sentences appearing in both $C_{classified_{i,t}}$ and $C_{judged_{j,t}}$. The average of size ratio between judged contexts and classified contexts, and the average of number of sentences in contexts is calculated. The results are shown in Table 9.

Table 9: Other evaluation metrics between contexts

Context Identification Phase	Size Ratio[a]	$\|C\|$[b]	C_{sim}[c]
I	0.2112	3.5361	0.9715
II	0.8104	13.1234	0.7596

[a] Average of size ratio between judged contexts and classified contexts

[b] Average number of sentences in classified contexts

[c] Equation 13

4.5 Scene Elements

A number of elements and attributes are defined for each scene. Elements of scene include objects, states, conversations and dialogues. Figure 10 depicts the structure.

4.5.1 Context

The context identified in previous steps has a unique identifier, location and time.

- **Location:** Locations extracted by the NER module of CoreNLP are run through Google Geocode API 2014 and complete latitude, longitude, and address is obtained.

```
<context>: context(scene)
    id - unique id for the scene
    location (optional) location info
    time (optional) temporal info
    <join>: arrival/addition of one or more objects to scene.
        id - unique id for arrival
    <state>: property identified for an entity
        id - unique id for state change
        what - what object
        event - what caused change
    <conv>: a conversation.
        id - unique id for conversation overall
    <dialogue>: an utterance in a conversation.
        id - unique id for the utterance
        by - who made the utterance
        to - who was it addressed to
```

Figure 10: Scene structure

- **Time:** TIPSemB and SUTime (CorNLP Temporal Tagger) (Chang and Manning 2012) are used for time extraction.

4.5.2 Join

Join tags mark the appearance of new objects in the current scene. As previously mentioned, an object is a tangible, physical entity. Figure 11 portrays the algorithm used to determine whether or not a given noun is an object. The algorithm uses a recursive function to assess the WordNet hypernym hierarchy of different synsets of a word. The order of the synsets is related to the relative frequency of different meanings. Hence, a coefficient is considered to reflect the relative importance of the ordering. The coefficient value obtained (0.65) is based on trial and error.

```
Function IsThing(array synsets){
  total = 0.0
  Coref = 0.0
  for each synset in synsets {
    if synset is an entity return -1;
    if synset is a physical entity return 1;
    hyp = synset.getHypernyms();
    total = total + coef * IsThing(hyp);
    coef = 0.65 * coef;
  }
  return total;
}
-------------------------------------------------------------------
if IsThing(Synset(nounTerm))>0 or the nounTerm is a proper noun then
  IsObject = True;
```

Figure 11: IsObject function algorithm

4.5.3 State

State tags mark the appearance of new states in the current scene. Dependency graphs are used to find the states, the objects the states modify, and the event causing the state. Each edge is checked, and based on the relation type, the governor part of speech (POS), and the dependent POS, distinct

conclusions are drawn. Table 10 shows a condensed set of rules for extracting states, modified objects and events.

4.5.4 Conversation

All dialogues within a certain proximity range are interpreted as conversation by a unique identifier.

Table 10: Rules for extracting states, modified objects and events

Relation	Gov[a] POS	Dep[b] POS	State	Object	Event
amod[27]	-	-	Dep	Gov	-
nsubj[28]	Noun	Adjective	Dep	Gov	-
nn[29]	Noun	Adjective	Dep	Gov	-
nn	Adjective	Noun	Gov	Dep	-
dobj[30]	-	Adjective	Dep	Gov	-
conj-and[31] conj-or[32]	Adjective	Adjective	Gov	Dep of dep	Event of dep
-	Verb	Adjective	Dep	Subj of dep	Gov

a Governor

b Dependent

[27] Adjectival modifier

[28] Nominal subject

[29] Noun compound modifier

[30] Direct object

[31] Conjunct with and

[32] Conjunct with or

4.4.5 Speech

Speech tags denote dialogue within scenes- usually one or more sentences, or parts of sentences, surrounded by quotation marks. Dialogue is generally at the beginning or end of a sentence if it is a part of a sentence. Dialogue also can be one or more complete sentences. The word “by” identifies who spoke (the speaker) and the word “to” specifies the audience. The following rules were applied to find the speaker and audience of a particular dialogue:

- If there is a word in the sentence which is a *person* and the *person* appears at the start of the sentence or at the end of the sentence, the *person* is the *speaker*.
- If the specific verbs *say*, *reply* and *whisper* appear in the sentence, the subject of the verb is the *speaker*.
- If there is a dialogue before this dialogue in the same conversation and the conversation has just two participants, the speaker of the previous dialogue is the audience of this dialogue and vice versa.
- If there is a dialogue after this dialogue in the same conversation and the conversation has just two participants, the speaker of the next dialogue is the audience of this dialogue and vice versa.

4.6 Scene Element Evaluation

Once the elements of scenes have been extracted, there are (1) sets of classified objects, states, and dialogues, and (2) sets of judged ones. The Jaccard index $\frac{|A \cap B|}{|A \cup B|}$, also known as the Jaccard

similarity coefficient, is used to evaluate the extracted elements and to measure the similarity and diversity of both the classified and judged sets. Table 11 shows the Jaccard similarities.

Table 11: Evaluation metrics for scenes elements

$J(\boldsymbol{Object_E}; \boldsymbol{Object_J})$	$J(\boldsymbol{State_E}; \boldsymbol{State_J})$	$J(\boldsymbol{Dialogue_E}; \boldsymbol{Dialogue_J})$
72.42	80.19	98.86

[a] Jaccard similarity between extracted objects and judged objects

[b] Jaccard similarity between extracted states and judged states

[c] Jaccard similarity between extracted dialogues and judged dialogues

False instances were investigated to find the error roots. In cases of joined objects, four reasons for the false instances were found:

- Coreference resolutions were the major cause of errors, accounting for 57%. This research uses CoreNLP to resolve coreferences and hence carries its intrinsic errors (e.g., "kid" is a coreference to a word "boy" which previously appeared in the scene, but because the CoreNLP coreference module doesn't recognize it, "kid" is extracted again as a joined object). The Average F-measure[33] of CoreNLP on CoNLL-2011 Shared Task data set is reported as 59.5%.

- The second most common source of errors was the IsObject methodology designed and implemented in this research. Some words such as *orphanage* and *arrow* represent both physical and abstract entities. *Orphanage* has two synsets in the WordNet: (1) "the condition of being a child without living parents" (2) "a public institution for the care of

[33] Avg F1 = (MUC + B cubed + CEAFE)/3.

orphans". Based on the first meaning orphanage is not an object, while the second meaning classifies orphanage as an object. The same problem occurs for *arrow:* (1) "a mark to indicate a direction or relation" (2) "a projectile with a straight thin shaft and an arrowhead on one end and stabilizing vanes on the other; intended to be shot from a bow". Although IsObject looks at all the synsets and their hierarchies and provides an overall estimate of being physical or abstract, the estimation could be incorrect because the word is not used in the winner senses (e.g. both words, orphanage and arrow, are used as their second sense in the corpus). This problem was the source of 29% of the errors.

  - A related source of errors was idioms. In idioms, a word can have both a figurative and a literal meaning. For example, the word *ball* in "he curled up in a ball" is used figuratively and does not mean a literal ball was present. 3% of the errors were caused by idioms.

- The next most common source of errors was incorrect POS tagging (e.g., tag VBZ[34] for the word *bullies* instead of NNS[35]). CoreNLP has high, but not perfect, accuracy in POS tagging. Although incorrect POS tagging has a relatively low impact on joined object accuracy, it causes significant errors in extracted states because of the common confusion between VBN[36] / VBG[37] and JJ[38]. This difficulty accounted for 11% of the errors.

[34] Verb, third person singular present

[35] Noun, plural

[36] Verb, past participle

[37] Verb, gerund or present participle

[38] Adjective

Figure 12 graphically depicts the different sources of errors.

Although wrong POS tag has low impact in joined object accuracy, it causes significant errors in extracted states because one of the common problems of POS taggers is confusion between VBN and VBG with JJ. This confusion lowers the state accuracy. In fact, the source of all errors in state extraction is from wrong POS tags. For example, in the sentence: “Suddenly, the barking of orders from the platoon leader interrupted Eddie's thoughts.”, the POS tagger specified the JJ tag for “interrupted”, while the correct POS is VBN. This error led our system to incorrectly extract “interrupted” as a state. In another example sentence: “Slowly, the man turned, and faced Eddie's questioning stare”, “questioning” is tagged as VBG by POS tagger, but the correct POS is JJ. This caused missing the state by our system.

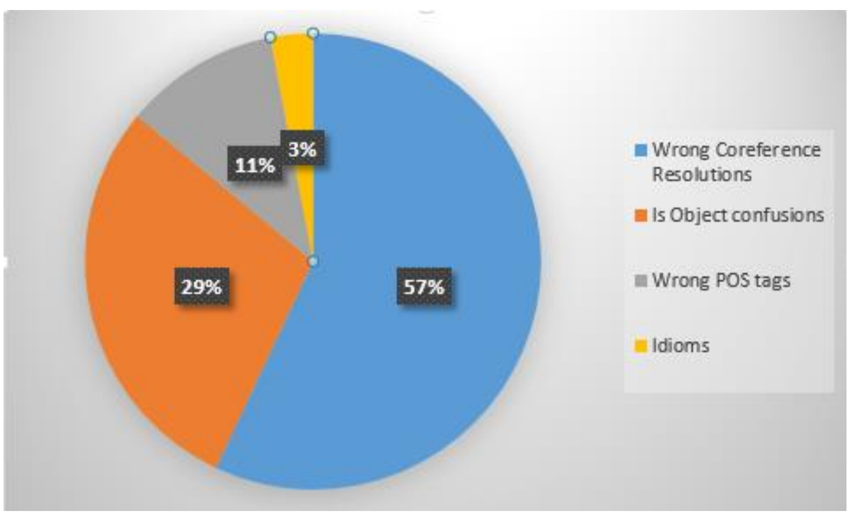

Figure 12: Error sources for object-object

For dialogue extraction our system focuses on sentences or parts of a sentences surrounded by quotation mark (“”), but quotation marks also are used to emphasize on a single or multi-word term. Our system limits the dialogue to be at the first or end of sentences or complete sentences,

so emphasize terms which usually appears in the middle of sentence will be filtered. This filter in very rare cases filters true dialogues, too (e.g. “Eddie looked at the medic's face and murmured a "thanks" as he accepted the offer.”, or “This smartly dressed dark-eyed man with a trim black beard and a splendid sapphire in his turban, asked Aladdin an unusual question: "Come here, boy," he ordered.”). Thus, the source of our slight errors for dialogue extraction is the small utterance which appears in the middle of a sentence.

4.7 Evaluation for Semantic Hierarchy

The semantic hierarchy proposed in this research could assist in advanced NLP problems such as textual entailment and word sense disambiguation. To evaluate the use of the semantic hierarchy an implemented WSD system was selected. WSD systems use adjacent words in a sentence, or adjacent sentences, to disambiguate a target word. The idea is that sentences in the same context, as opposed to any adjacent sentences of a different context, are more useful to WSD systems.

Babelfy (Moro, Cecconi, and Navigli 2014) is a high-accuracy word sense disambiguation system. It is a graph-based system, providing better results on a document as a whole, rather than sentence by sentence. In the experiment, short stories were fed to Babelfy in the following four ways: (1) the whole story in one query, (2) each context in one query, and (3) each scene in one query (4) five consecutive sentences in one query. The Babelfy disambiguations were reviewed to compare its accuracy on the four inputs. A total of 11,176 ambiguous words in twenty short stories including 28,154 words, 181 scenes, 374 contexts were disambiguated by Babelfy separately in four different settings (whole story WSD, scene WSD, context WSD, five consecutive sentences WSD). The four different disambiguations performed by Babelfy were reviewed. In case of any

difference between the disambiguations, it was determined which disambiguation was the correct disambiguation. Then, it was calculated to what extent context WSD or scene WSD improved the accuracy of whole story WSD. Table 12 shows the results of the experiment.

Table 12: Evaluation by word sense disambiguation

Input	Accuracy Improvement
Contexts WSD	2.28 %
Scenes WSD	2.07 %
Five sentence WSD	-3.42 %

Our experiment shows 2.28% growth of the accuracy of Babelfy which is a significant improvement in WSD system accuracies. The accuracy of Babelfy on AIDA-CoNLL[39] is reported as 82.1% (Moro, Raganato, and Navigli 2014) and the top seven WSD systems had accuracy between 80.07% and 82.3%. However, there is no significant improvement of accuracy from context setting to scene setting. It appears that while a scene provides more information about the topic, the information elements are not as closely topically related. Conversely, context provides more topical information. Thus, providing context around target word helped disambiguation significantly. To insure that this improvement was not just an artifact of shorter segments, stories were divided up into sets of 5 consecutive sentences (based only on sequence, not context/scene). These sets were tested in Babelfy and accuracy decreased by 3.42%. This accuracy reduction

[39] consists of 1392 English articles, for a total of roughly 35K named entity mentions annotated with YAGO concepts separated in development, training and test sets.

shows that a blind text segmentation and using fixed window of sentences doesn’t help the WSD system more than providing the whole story.

CHAPTER 5- CONTEXT AWARE TEXT REPRESENTATION

The *syntactic* hierarchy of text includes words, clauses, sentences, and paragraphs. In this chapter an innovative *semantic* hierarchical representation of a text is generated to summarization, question answering, and textual entailment tasks relating to stories. This semantic hierarchy categorizes a raw text to its scenes and contexts. Also, in this structure, concise and important semantic details are provided. The semantic hierarchy proposed in this chapter is shown in Figure 13.

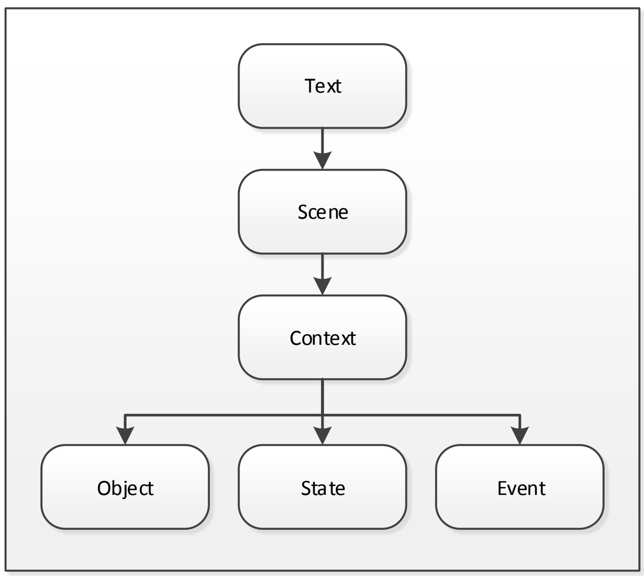

Figure 13: Hierarchical semantic structure

5.1 Structure

A graph representing text is specified as follows:

$$G = \{V, E\} \quad \textbf{(Eq. 14)}$$

where V represents context nodes and E links between nodes (the node structure). V is the union of the different node types:

$$V = V_t \cup V_{sc} \cup V_c \cup V_o \cup V_s \cup V_e \qquad \textbf{(Eq. 15)}$$

where V_t is a node representing texts, V_{sc} is a node representing scenes, V_c is a node representing contexts, V_o is a node representing objects, V_s is a node representing states, and V_e is a node representing events.

The edges (E) are links between the nodes and provide hierarchical structure: text node ← scene node ← context nodes. Context nodes are connected to their objects, states, and events. The structure is a tree which connects the above nodes.

$$E = \{(v_t, v_{sc}) \cup (v_{sc}, v_c) \cup (v_c, v_o) \cup (v_c, v_s) \cup (v_c, v_e) \mid v_t \in V_t, v_{sc} \in V_{sc}, v_c \in V_c, v_o \in V_o, v_s \in V_s, v_e \in V_e\} \qquad \textbf{(Eq. 16)}$$

Figure 14 depicts an abstract view of the proposed structure. Each node and edge contains different information from lexical level to semantic level.

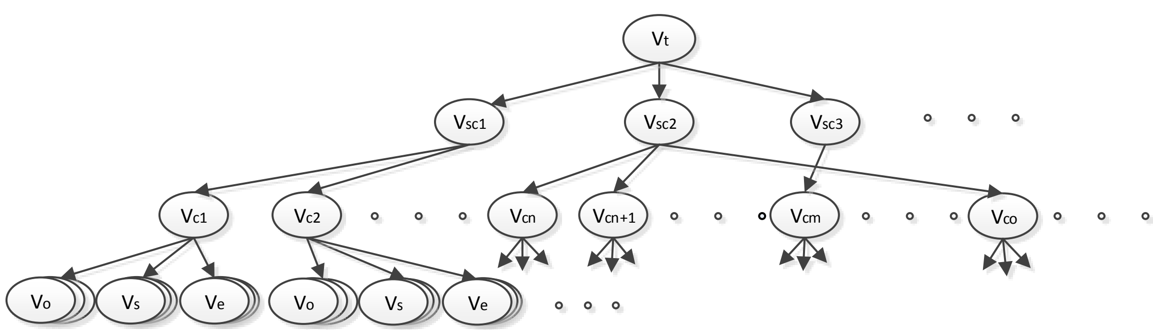

Figure 14: Text representation structure

5.1.1 Scene nodes

A scene node (v_{sc}) represents a set of contexts occurring in the same period of time and in the same location (scene).

$$V_{sc} = \{id, Time, Locs, Contexts\} \quad \textbf{(Eq. 17)}$$

- **Identity (id):** Identity number of the scenes.
- **Time (T):** Time of the scene based on Timex3 representation.
- **Locations (Locs):** Locations of the scene. The locations are represented by their Coordinate (Latitude, Longitude) of North-East and South-West of the location.
- **Contexts:** contexts in the scene.

5.1.2 Context nodes

A context node (v_c) represents a set of adjacent sentences with cohesive and consistent semantical structure.

$$V_c = \{id, Objs, States, Events\} \quad \textbf{(Eq. 18)}$$

- **Identity (id):** Identity number of the contexts.
- **Objects (Objs):** Tangible objects mentioned in the context.
- **States:** States of the objects in the context.
- **Events:** Events occurring in the context.

5.1.3 Object nodes

Each object node (v_o) contains folowing information.

$$V_o = \{id, Synset\} \quad \textbf{(Eq. 19)}$$

- **Identity (id):** Identity number of the objects.
- **Synset (S):** Synset identifier of the object in WordNet.

5.1.4 State nodes

Each state node (v_s) represents a state includes various information.

$$V_s = \{id, Modifier, Polarity, Modified - Object\} \quad \textbf{(Eq. 20)}$$

- **Identity (id):** Identity number of the states.
- **Modifier:** Synset identifier of the modifier.
- **Polarity:** Polarity of the modifier.
- **Modified Object:** Synset identifier and object-id of object modified by the modifier.

  Object-id may be blank if the modified object is not a tangible object.

5.1.5 Event nodes

$$V_e = \{id, Synset, Polarity, Subjects, Objects, Adv, Deps, Time, Loc\} \quad \textbf{(Eq. 21)}$$

- **Identity (id):** Identity number of the events.
- **Synset:** Synset identifier of the event.

- **Polarity:** The event occurred or is negated (did not occur, such as "He did not *jump* through the hoop").

- **Subjects:** Synset identifier and object-id of subjects of the event. Object-id may be blank if subject is not a tangible object.

- **Objects:** Synset identifier and object-id of objects of the event. Object-id may be blank if object is not a tangible object.

- **Adverbs:** Synset identifier of the adverbs modifying the event.

- **Dependents:** Synset identifier and relation of the dependent of the event.

  e.g. "The mass of helmets lurched backward as the landing craft plunged into the dark water."

$$V_e = \{id = 2, Synset = \text{"plung\#v\#1"}, Polarity = \text{"POS"}, Subjects, Deps\}$$

$$Deps = \{relation = \text{"}prep{:}\,into\text{"}, Synset = \text{"water\#n\#1"}\}$$

- **Time:** Time of the event based on Timex3 representation.

- **Location:** Location of the event.

5.2 Extraction and Loading of Representation from Stories

In the conversion of a natural language sentence to the proposed representation, the dependency graph plays an intermediate role. From the root of the dependency graph, all links are traversed by breadth-first order, and the semantic data are extracted. Temporal extraction tools, Google GeoCode API, Babelfy Word Sense Disambiguition are other preprocessing tools used to provide

the representation. At last, the scenes of the document are sorted based on their time information. Thus, the text representation offers a linear narrative of the story. The extraction process used all the above tools along with the scene and the context identifiers discussed in Chapter 4 to provide the semantics needed.

5.2.1 Extracting scene nodes' semantic data

The scenes are identified in chapter 4. Our scene identifier categorizes the text into different scene. Each scene node includes identifier, Time, and location.

- **Identity (id):** a unique integer starting with 1 is assigned to each scene.
- **Time (T):** Time expressions are extracted by TIPSemB[40] (Llorens, Saquete, and Navarro 2010), a temporal information processing system that extracts times, events and temporal relations from raw text and SUTime[41], Stanford's temporal expression recognizer (Chang and Manning 2012). Our system uses both systems outputs. To extract information from the TIPSemB output file, the sentences in the file are first aligned to our sentences (as a different tagging structure is used) and then the <TIMEX3> tags in the sentence are extracted. The first appearance of time in the scene is used as the time of the scene. Figure 15 shows a part of TIPSemB output file which include Timex3 tag.

[40] http://gplsi.dlsi.ua.es/demos/TIMEE/

[41] http://nlp.stanford.edu:8080/sutime/process

```
On the other hand, Eddie <EVENT class="STATE" eid="e56">had</EVENT> the advantage of a
loving, structured family life. His dad was a kind and forgiving man. It was his
mother's Irish temperament which had <EVENT class="ASPECTUAL" eid="e57">kept</EVENT>
him on his toes. His sister, Tess, had the same fire and beauty as their mother. Vince
had been <EVENT class="OCCURRENCE" eid="e58">smitten</EVENT> with Tess for <TIMEX3
type="DURATION" value="PXY" tid="t2">years</TIMEX3>. In fact, Eddie was sure that his
friend was in love.
```

Figure 15: Part of TIPSemB output file

- **Locations (Locs):** The locations are tagged by the Named-Entity Recognizer embedded in CoreNLP. The locations recognized by NER are fed to the Google Geocode API[42]. The information extracts are the location name, hierarchy and GIS coordinates of the North-East and South-West corners.

- **Contexts:** The extraction of contexts in the scene are discussed in Chapter 4. Our context identifier categorizes each scene into different contexts.

5.2.2 Extracting context nodes' semantic data

Each context node includes an identifier, objects, states, and events.

- **Identity (id):** a unique integer starting with 1 is assigned to each context.

- **Objects (Objs):** Extraction of tangible objects is discussed in Section 4.5.2. The algorithm recursively looks up the hypernyms of the nouns and checks if they are 'Physical entity' or not. Since the hypernyms are usually more than one, this algorithm uses a weighted formula and a threshold to decide if the object is tangible or not.

[42] https://developers.google.com/maps/documentation/javascript/tutorial

- **States:** States are usually adjectives. The states are extracted based on the information from the dependency parser output and POS tagger. More explanation about state extraction comes in the State node below.

- **Events:** Events are extracted from TIPSemB output file. Events are tagged with <EVENT> in the file. An example of this tag is shown in Figure 15.

5.2.3 Extracting object nodes' semantic data

The semantic data for the extracted objects include an identifier and synset identifier. The synset is provided by using Babelfy WSD API.

- **Identity (id):** a unique integer starting with 1 is assigned to each object.

- **Synset (S):** The synset is provided by using Babelfy WSD API. Since Babelfy is internet-based API and has limitation on the number of queries for each day, all the files went through Babelfy API and the synset identifier available for each word in the files is stored in a file. Then, the file is used to provide any synset identifier our representation needs.

5.2.4 Extracting state nodes' semantic data

State node includes identifier, modifier, and modified object. Our system traverses the dependency graph for each sentence of the texts and based on the dependency relations and POS tag of governor (parent) and dependent (child) extracts State relationship. Following rules and examples show how the *modifier* and *modified objects* are extracted

- $(X\ is\ governor) \wedge (Y\ is\ dependent) \wedge (Relation\ is\ amod) \Rightarrow$

  $(X\ is\ modified\ object)\ \wedge (Y\ is\ modifier)$

e.g., in sentence: "Eddie had the advantage of a loving, structured family life"

$(\text{"life "}\ is\ governor) \wedge (\text{"}structured\text{"}\ is\ dependent) \wedge (Relation\ is\ amod)$

$$\Rightarrow (\text{"life" is modified object) and ("structured" is modifier})$$

- $(X\ is\ governor) \wedge\ (Y\ is\ dependent) \wedge\ (Relation\ is\ acomp) \wedge$

  $(Z\ is\ dependent\ of\ X) \wedge (Relation\ between\ X\ and\ Z\ is\ nsubj) \Rightarrow$

  $(Z\ is\ modified\ object)\ and\ (X\ is\ modifier)$

e.g., in sentence: "Eddie lay against the cliff, feeling weak and helpless."

$(\text{"}feeling\text{"}\ is\ governor)\ \wedge\ (\text{"}weak\text{"}\ is\ dependent)\ \wedge\ (Relation\ is\ acomp)\ \wedge$

$(\text{"Eddie" is dependent of "feeling"})\ \wedge$

$(Relation\ between\ \text{"}feeling\text{"}\ and\ \text{"}Eddie\text{"}\ is\ nsubj) \Rightarrow$

$(\text{"}Eddie\text{"}\ is\ modified\ object)\ and\ (\text{"}weak\text{"}\ is\ modifier)$

- $(X\ is\ governor) \wedge\ (Y\ is\ dependent) \wedge\ (Relation\ is\ conj - and\ or\ conj - or) \wedge$

  $((\ X\ is\ an\ adjective) \vee\ (\ Y\ is\ an\ adjective))\ \wedge$

  $(X\ is\ Modifier) and\ (Z\ is\ modified\ object) \Rightarrow$

  $(Y\ is\ Modifier)\ and\ (Z\ is\ modified\ object)$

e.g., in sentence: "The black and white stripes painted on their lower bodies could be seen."

$(\text{"black"}\ is\ governor) \wedge\ (\text{"white"}\ is\ dependent) \wedge\ (Relation\ is\ \ conj - and) \wedge$

$(\ \text{"white"}\ is\ an\ adjective) \wedge$

$("black" \ is \ Modifier) \ and \ ("stripe" \ is \ modified \ object) \Rightarrow$

$("white" \ is \ Modifier) \ and \ ("stripe" \ is \ modified \ object)$

- $(X \ is \ governor) \ \wedge \ (Y \ is \ dependent) \ \wedge \ (Relation \ is \ nn) \ \wedge \ (Y \ is \ an \ adjecive) \ \Rightarrow$

  $(X \ is \ Modifier) \ and \ (Y \ is \ modified \ object)$

e.g., in sentence: "They are always ready to help kind or polite people" $("people" \ is \ governor) \ \wedge \ ("polite" \ is \ dependent) \ \wedge \ (Relation \ is \ nn) \ \wedge$

$("polite" \ is \ Modifier) \ and \ ("people" \ is \ modified \ object)$

Modifier objects are represented with synset identifier and object-id. Object-id is the identifier assigned to the tangible object. Thus, if the modified object is not tangible the object-id will be blank.

5.2.5 Extracting event nodes' semantic data

Event nodes include identifier, event, subjects, objects, adverbs, dependents, time and location.

- **Identity (id):** a unique integer starting with 1 is assigned to each event.

- **Event:** Events which are tagged in TIPSemB output file are extracted and synset identifiers of them are represented.

- **Subject:** Our system finds subjects of the event by working on dependency graph. When the dependency relation is nsubj, agent, or xcomp[43], a subject is extracted by following rules.

[43] Open clausal complement (e.g., "He says that you like to swim" xcomp(like, swim))

- $(X \ is \ governor) \wedge (Y \ is \ dependent) \wedge (Relation \ is \ nsubj) \Rightarrow (Y \ is \ Subject)$

e.g., for the sentence: "Eddie felt a sharp pain in his side."

$$("felt" \ is \ governor) \wedge ("Eddie" \ is \ dependent) \wedge (Relation \ is \ nsubj) \Rightarrow ("Eddie" \ is \ subject)$$

- $(X \ is \ governor) \wedge (Y \ is \ dependent) \wedge (Relation \ is \ agent) \Rightarrow (Y \ is \ Subject)$

e.g., in sentence: "Men were distracted by an explosion."

$$("distracted" \ is \ governor) \wedge ("explosion" \ is \ dependent) \wedge (Relation \ is \ agent) \Rightarrow ("explosion" \ is \ subject)$$

- $(X \ is \ governor) \wedge (Y \ is \ dependent) \wedge (Relation \ is \ xcomp) \wedge (Z \ is \ dependent \ of \ X) \wedge (Relation \ between \ X \ and \ Z \ is \ nsubj) \Rightarrow (Z \ is \ Subject)$

  e.g., in sentence: "Sea spray glistened on the surface of everything it touched, catching the light of the artillery fire."

$$\begin{aligned} &(\text{"touched" is governor}) \wedge (\text{"catching" is dependent}) \\ &\quad \wedge (\text{Relation is xcomp}) \wedge (\text{"it" is dependent of "touched"} \\ &\quad \wedge (Relation \ between \ "touched" \ and \ "it" \ is \ nsubj) \\ &\quad \Rightarrow Subject("catching", "it") \end{aligned}$$

  In this sentence, "it" refers to "sea spray". Our system keeps reference of each word if exists, and always uses the referents in FOL sentences if available.

Therefore, our system produces $Subject("touched", "sea\ spray")$ in this example.

- **Object:** If one of the dependency relations dobj, iobj, and nsubjpass[44] exists, an object is extracted. Following rules elaborates generating of Object element.

  - $(X\ is\ governor) \wedge (Y\ is\ dependent) \wedge ((Relation\ is\ \text{dobj}) \vee (Relation\ is\ \text{iobj}) \vee (Relation\ is\ \text{pobj}) \vee (Relation\ is\ \text{nsubjpass})) \Rightarrow (Y\ is\ Object)$

  e.g., in sentence: "The roar of airplanes filled the sky."

$$(\text{"filled "}\ is\ governor) \wedge (\text{"sky"}\ is\ dependent) \wedge (Relation\ is\ \text{dobj}) \Rightarrow (\text{"sky"}\ is\ Object)$$

  e.g., in sentence: "Men were distracted by an explosion."

$$("distracted"\ is\ governor) \wedge ("men"\ is\ dependent) \wedge (Relation\ is\ \text{nsubjpass}) \Rightarrow ("men"\ is\ Object)$$

  e.g., in sentence: "Give me your gear."

$$("Give"\ is\ governor) \wedge ("me"\ is\ dependent) \wedge (Relation\ is\ \text{iobj}) \Rightarrow ("me"\ is\ Object)$$

[44] Passive nominal subject

In this sentence, "me" refers to "Eddie Hagen". Therefore, our system produces $("Eddie\ Hagan"\ \ is\ Object)$ in this example.

- **Adverbs:** adverbs are connected with *advmod* relation to the events.
  - $(X\ is\ governor) \wedge (Y\ is\ dependent) \wedge (Relation\ is\ advmod) \Rightarrow (Y\ is\ Adverb)$

    e.g., in sentence: "The mass of helmets lurched backward as the landing craft plunged into the dark water."

$$("lurched"\ is\ governor) \wedge ("backward"\ is\ dependent) \wedge (Relation\ is\ advmod) \Rightarrow ("backward"\ is\ Adverb)$$

- **Dependents:** There are some other nouns in sentences which are not subject or object, but are important in the sentences. They are usually connected to the verb by prepositions. We call them dependents.
  - $(X\ is\ governor) \wedge (Y\ is\ dependent) \wedge (Relation\ starts\ with\ prep) \Rightarrow (Y\ is\ Dependent)$

  e.g., in sentence: "The mass of helmets lurched backward as the landing craft plunged into the dark water."

$$("plunged"\ is\ governor) \wedge ("water"\ is\ dependent) \wedge (Relation\ is\ prep\_into) \Rightarrow ("water"\ is\ Dependent)$$

- **Time:** times connected to the events are connected by *tmod* relation in dependency graph.

- $(X\ is\ governor) \wedge (Y\ is\ dependent) \wedge (Relation\ is\ tmod) \Rightarrow (Y\ is\ time)$

  e.g., in sentence: “They're blowing us all to hell now.”

  $$("blowing"\ is\ governor) \wedge ("now"\ is\ dependent) \wedge (Relation\ is\ tmod) \Rightarrow ("now"\ is\ Time)$$

For representing time, Timex3 standard representation is used. For example, Timex3 representation of “now” is *PRESENT_REF*.

- **Location:** if a word is dependent of the event and NER recognizes it as a location, the location will be assigned to the event.

  e.g., “San Diego Road” in sentence: “The boy, now about twelve jumped up and ran across San Diego Road, placed his fingers through the chain-link fence and just stood there looking at us.”

5.3 A Short Example of the Text Representation

Dependency graph for the sentence “Alex gave Mary Watson an interesting book” is shown in Figure 16.

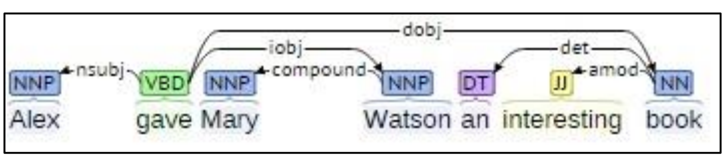

Figure 16: Dependency graph (Manning et al. 2014)

Our system produces following node based on the dependency graph and all the rules embedded in:

$V_{o1} = \{id = 1, Synset = "Alex"\}$

$V_{o2} = \{id = 2, Synset = \text{"Mary Watson"}\}$

$V_{o3} = \{id = 3, Synset = \text{"book\#n\#1"}\}$

$$V_s = \{id = 1, Modifier = \text{"interesting"}, Modified - Object = \{\text{object\_id} = \text{"3" synset} = \text{"book\#n\#1"}\}\}$$

$$V_e = \{id = 1, Synset = \text{"give\#v\#1"}, Polarity = \text{"POS"}, Subjects = \{object\_id = 1, \text{synset} = \text{"Alex"}\}, Objects = \{object\_id = 2, \text{synset} = \text{"Mary Watson"}; object\_id = 3, \text{synset} = \text{"book\#n\#1"}\}\}$$

5.4 File Structure

The above sections detail the data representation structure in memory. To load and save data to file, a file structure is designed and implemented. This structure summarizes states explained and events happened in the scene-context hierarchy. Introducing objects in the beginning of each context helps applications to query faster, because they can immediately find the context in which they want to concentrate based on their query. This file will be produced in xml, which is easy to retrieve. Figure 17 shows the structure of the file, and Figure 18 depicts a part of a file is created. Table 13 shows statistics about memory and time to produce these files for twenty stories. The total time includes all preprocessing of the stories, information extraction, and saving to file the new text representation. Table 14 shows the number of different elements in the files.

The average time spent for whole process is for analyzing the text, finding the semantic hierarchy, and extracting deep semantic details. The spent time is reasonable comparing with other NLP tools such as Stanford parser or TIPSem temporal system. The representation provided is developed by the processing on the unstructured file, the CoreNLP output, and TIPSem output on the files with the average size of 7.8 kB, 1.2 MB, and 55.5 kB. Our representation also includes

other information extracted from Google geocode API and Babelfy API. By all these different sources of information our concise structure has allocated just 40.59 kB memory in average.

```
<document>
    <scene id="">
        [<time expr=""  timex3=""/>]
        [<location address="" coordinate=""/>]
        <context id ="">
            <objects>
                <object_id="" synset=""/>
                ...
            </objects>
            <states>
                <state id="" modifier="" polarity="">
                    <modifier_object [object_id=""] synset=""/>
                </state>
                ...
            </states>
            <events>
                <event id="" synset="" polarity="">
                    [<subject [object_id=""] synset=""/>]
                    [<object [object_id=""] synset=""/>]
                    [<adverb synset=""/>]
                    [<dependent relation="" synset=""/>]
                    [<time expr=""  timex3=""/>]
                    [<location address="" coordinate=""/>]
                </event>
                ...
            </events>
        </context id ="">
        ...
    </scene>
    ...
</document>
```

Figure 17: Text representation file structure

Table 13: Statistics of the text representation structure

Number of files	**Average total time (seconds/file)**	**Average time spent to save (milliseconds/file)**	**Average size (kB/file)**
20	34.35	4.65	40.59

Table 14: Number of different nodes

Number of files	Number of scene nodes	Number of context nodes	Number of tangible object nodes	Number of state nodes	Number of event nodes
20	181	374	2542	1372	5963

For theses sentences in a story in our corpus: “Eddie looked at the medic's face and murmured a "thanks" as he accepted the offer. Still trembling, he brought the cigarette to his lips and inhaled slowly. Lying back, he could only think of the fact that he was on a beach.” Our system identifies these sentences as a context and produces the nodes in the Figure 18.


```
▼<context id="23">
  ▼<objects>
     <object id="167" synset="Eddie Hagen"/>
     <object id="168" synset="medic#n#2"/>
     <object id="169" synset="cigarette#n#1"/>
     <object id="170" synset="beach#n#1"/>
   </objects>
   <states/>
  ▼<events>
    ▼<event id="197" polarity="POS" synset="look#v#1">
       <subject synset="Eddie Hagen"/>
       <dependent relation="prep:at" synset="face#n#1"/>
     </event>
    ▼<event id="198" polarity="POS" synset="murmur#v#1">
       <subject synset="Eddie Hagen"/>
       <object synset="thanks#n#1"/>
     </event>
    ▼<event id="199" polarity="POS" synset="accept#v#1">
       <subject synset="Eddie Hagen"/>
       <object synset="offering#n#3"/>
     </event>
    ▼<event id="200" polarity="POS" synset="tremble#v#1">
       <adverb synset="Still"/>
     </event>
    ▼<event id="201" polarity="POS" synset="take#v#7">
       <subject synset="Eddie Hagen"/>
       <object synset="cigarette#n#1"/>
       <dependent relation="prep:to" synset="lip#n#1"/>
     </event>
    ▼<event id="202" polarity="POS" synset="inhale#v#1">
       <subject synset="Eddie Hagen"/>
       <adverb synset="slowly#r#1"/>
     </event>
    ▼<event id="203" polarity="POS" synset="lie#v#1">
       <adverb synset="back"/>
     </event>
    ▼<event id="204" polarity="POS" synset="think#v#1">
       <subject synset="Eddie Hagen"/>
       <adverb synset="only"/>
       <dependent relation="prep:of" synset="fact#n#1"/>
     </event>
   </events>
 </context>
```


Figure 18: The text representation nodes for the example text

5.5 Equivalent First-Order Logic Representation

First-order logic (FOL) enables use of first-order reasoning tools and recent logical statistical reasoning tools like Markov logic network (MLN) (Richardson and Domingos 2006) which can handle uncertainty. Based on the text representation provided, equivalent FOL formulas are generated.

- **Scene node to FOL clauses**

  scId = "sc" + Id

  FOL: Location(locId, address, coordinate)

  Time(tId, timeExpr, timex3)

  Scene(scId) ^ SceneTime(scId, tId) ^ SceneLocation(scId, locId) ^ isa(cId, scId)

  Scene identifier is built by attaching "sc" to the identifier of the scene. Then FOL is created by the new identifier. *"isa"* provides hierarchical link between the scene and its contexts.

- **Context node to FOL clauses**

  cid = "c" + Id

  FOL: Context(cId) ^ ist(cId, objId) ^ ist(cId, stId) ^ ist(cId, eId)

  Context identifier is built by attaching "c" to the identifier of the context. Then equivalent FOL is created by the new identifier. Relation $ist(c, p)$ that expresses "p is true in the context c" which is proposed by McCarthy is used to express objects, states, and events in the contexts.

- **Object node to FOL clauses**

  objId = "obj" + Id

FOL: Thing(objId, synset)

Object identifier is built by attaching “obj” to the identifier of the object. The term “Thing” is chosen for the object to prevent confusion between the object of event and this object.

- **State node to FOL clauses**

stId = "st" + Id

FOL: Adjective(stId) ^ State(objId, stId)

State identifier is built by attaching “st” to the identifier of the state. Then the adjective is introduced with the id and state relation is made between modified object identifier and adjective identifier.

- **Event node to FOL clauses**

eId = "e" + Id

FOL: Location(locId, address, coordinate)

Time(tId, timeExpr, timex3)

Event(eId, synset) ^ Subject(eId, objId) ^ Object(eId, objId) ^ Adverb(eId, synset) ^ Dependent-Relation(eId, objId) ^ EventTime(eId, time-expr) ^ EventLocation(eId, coordination)

Event identifier is built by attaching “e” to the identifier of the event. Then equivalent FOL is created by the new identifier.

By applying the above notation rules to “Alex gave Mary Watson an interesting book”, the logical forms of this sentence in scene *sc1* and context *c*1 are shown in Figure 19.

$Scene(sc1)$

$Context(c1)$

$isa(c1, sc1)$

$ist(c1, obj1)$

$ist(c1, obj2)$

$ist(c1, obj3)$

$ist(c1, st1)$

$Thing(obj1, Alex)$

$Thing(obj2, Mary\ Watson)$

$Thing(obj3, book)$

$Adjective(st1)$

$State(obj3, st1)$

$e1(give)$^ $subject(e1, obj1)$ ^ $object(e1, obj2)$ ^ $object(e1, obj3)$

Figure 19: First-order-logic representation

CHAPTER 6- KNOWLEDGE GENERATION AND REASONING

In the previous chapters, semantic hierarchy of texts is found and then information in the unstructured text are organized into a structured format suitable for sematic oriented NLP applications. In this chapter, new knowledge based on the explicit facts mentioned in the text is produced. To produce this extra knowledge, the reasoning process works on the extracted structural data in the previous chapters and VerbOcean semantic network of verbs. The new knowledge is saved in an extended Textual Entailment structure. This structure can serve as an information source for another application, also it can be queried directly by XQuery to answer questions about the text. Figure 20 shows an overview of this process.

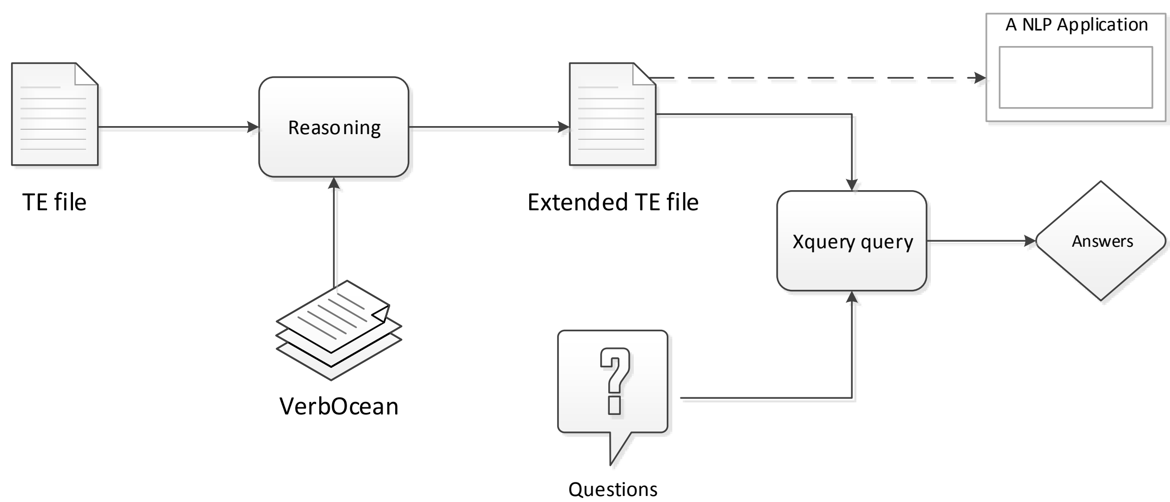

Figure 20: Knowledge generation overview

6.1 Knowledge Generation

In the knowledge generation step, our reasoning process works on facts and by employing inference rules produces new knowledge about the story. The facts are extracted from the text

entailment file produced in previous chapters and also VerbOcean. The inference rules are produced based on WordNet and VerbOcean.

6.1.1 Facts

The facts are categorized to two groups: (1) story facts (2) VerbOcean verb connections.

- **Story facts:** In the previous chapter, a first-order logic representation and also xml representation of the story are produced for each scene and context. Since, our inference rules are verb entailment rules, the story facts include the verbs in the story. The verbs are extracted and tagged in the event structure of the text entailment file.

- **VerbOcean verb connections:** These facts are produced from VerbOcean. VerbOcean groups the relationship between two verbs into different categories. A numeric weight is assigned to each verb pair in VerbOcean indicating the strength of the relation. Therefore, weight normalization process is applied to VerbOcean relations. Then, all relations whose first argument is a verb within the context are extracted. Figure 21 depicts a portion of VerbOcean evidences.

6.1.2 Inference rules

In order to provide inference rules, VerbOcean relations are used. For any verb argument in event nodes, if there exists a VerbOcean relation between the verb argument and another verb, then the second verb is entailed. Figure 22 shows the inference rules used for reasoning. Synonym (x,y) means x and y are two different word forms of one synset. The synonyms are extracted based on WordNet.

```
0.3822876293042862 happens-before("accept","abandon")
0.26083238574522005 similar("accept","receive")
0.2371841665946477 stronger-than("accept","discuss")
0.274416739006976 similar("accept","propose")
0.23471378377963545 similar("accept","consider")
0.22838539605100167 similar("accept","reconsider")
0.28953302071431775 similar("accept","negotiate")
0.3530418429052019 happens-before("accept","implement")
0.24357641113884246 similar("accept","submit")
0.3311614984269622 happens-before("accept","back")
0.3361357594915892 happens-before("accept","cancel")
0.3049932241684607 stronger-than("accept","concede")
0.2908049172205322 stronger-than("accept","contemplate")
0.33024060080677875 happens-before("accept","defend")
0.35755298989095685 happens-before("accept","dismiss")
0.40687944055516834 happens-before("accept","embrace")
0.34172790925050905 happens-before("accept","promise")
0.36771870307189425 happens-before("accept","return")
0.4693908435579471 happens-before("accept","spurn")
0.37019022004383445 happens-before("accept","violate")
```

Figure 21: VerbOcean verb connections

```
Event(x) ^ happens-before(x,y) -> EntailedEvent(y).
Event(x) ^ stronger-than(x,y) -> EntailedEvent(y).
Event(x) ^ similar(x,y) -> EntailedEvent(y).
Event(x) ^ can-result-in(x,y) -> EntailedEvent(y).

Event(x) ^ Synonym(x,y) ^ happens-before(y,z) -> EntailedEvent(z).
Event(x) ^ Synonym(x,y) ^ stronger-than(y,z) -> EntailedEvent(z).
Event(x) ^ Synonym(x,y) ^ similar(y,z) -> EntailedEvent(z).
Event(x) ^ Synonym(x,y) ^ can-result-in(y,z) -> EntailedEvent(z).
```

Figure 22: Inference rules

6.1.3 Inference results

The results are saved in a new xml file. The file structure is expanded to include the entailments. In each event node all the entailed verbs are listed with <EntailedEvent> tag. A normalized VerbOcean weight is associated with each entailed event. In Figure 23 the modified file structure is shown. Figure 24 shows a part of the extended file including entailed events. This file part

represent this part of a story in our corpus: “Eddie looked at the medic's face and murmured a "thanks" as he accepted the offer. Still trembling, he brought the cigarette to his lips and inhaled slowly. Lying back, he could only think of the fact that he was on a beach.” Table 15 shows statistics about the time to produce extended files for twenty stories. In Table 16, the number of entailed events is shown.

```
<document>
    <scene id="">
        [<time expr=""  timex3=""/>]
        [<location address="" coordinate=""/>]
        <context id ="">
            <objects>
                <object_id="" synset=""/>
                ...
            </objects>
            <states>
                <state id="" modifier="" polarity="">
                    <modifier_object [object_id=""] synset=""/>
                </state>
                ...
            </states>
            <events>
                <event id="" synset="" polarity="">
                    [<subject [object_id=""] synset=""/>]
                    [<object [object_id=""] synset=""/>]
                    [<adverb synset=""/>]
                    [<dependent relation="" synset=""/>]
                    [<entailedEvent verb="" verbOceanWeight="">]
                    ...
                    [<time expr=""  timex3=""/>]
                    [<location address="" coordinate=""/>]
                </event>
                ...
            </events>
        </context id ="">
        ...
    </scene>
    ...
</document>
```

Figure 23: Extended text entailment file structure

```
▼<context id="23">
  ▼<objects>
      <object id="167" synset="Eddie Hagen"/>
      <object id="168" synset="medic#n#2"/>
      <object id="169" synset="cigarette#n#1"/>
      <object id="170" synset="beach#n#1"/>
    </objects>
    <states/>
  ▼<events>
    ▼<event id="197" polarity="POS" synset="look#v#1">
        <subject synset="Eddie Hagen"/>
        <dependent relation="prep:at" synset="face#n#1"/>
      </event>
    ▼<event id="198" polarity="POS" synset="murmur#v#1">
        <subject synset="Eddie Hagen"/>
        <object synset="thanks#n#1"/>
      </event>
    ▼<event id="199" polarity="POS" synset="accept#v#1">
        <subject synset="Eddie Hagen"/>
        <object synset="offering#n#3"/>
        <entailedEvent verb="abandon" verbOceanFactor="0.3972246523833035"/>
        <entailedEvent verb="receive" verbOceanFactor="0.38717688618200086"/>
        <entailedEvent verb="propose" verbOceanFactor="0.4073413591697243"/>
        <entailedEvent verb="consider" verbOceanFactor="0.34840670451314926"/>
        <entailedEvent verb="reconsider" verbOceanFactor="0.33901291145204465"/>
        <entailedEvent verb="negotiate" verbOceanFactor="0.42977981084196176"/>
        <entailedEvent verb="implement" verbOceanFactor="0.36683615313420 59"/>
        <entailedEvent verb="submit" verbOceanFactor="0.36156229657862604"/>
        <entailedEvent verb="back" verbOceanFactor="0.3441008837633059"/>
        <entailedEvent verb="cancel" verbOceanFactor="0.3492695028103207"/>
        <entailedEvent verb="concede" verbOceanFactor="0.4527287763855358"/>
        <entailedEvent verb="contemplate" verbOceanFactor="0.4316678008145831"/>
        <entailedEvent verb="defend" verbOceanFactor="0.34314400415481927"/>
        <entailedEvent verb="discuss" verbOceanFactor="0.35207371512328206"/>
        <entailedEvent verb="dismiss" verbOceanFactor="0.37152356296885736"/>
        <entailedEvent verb="embrace" verbOceanFactor="0.42277733294841835"/>
        <entailedEvent verb="promise" verbOceanFactor="0.3550801531525901"/>
        <entailedEvent verb="return" verbOceanFactor="0.3820864783628893"/>
        <entailedEvent verb="spurn" verbOceanFactor="0.48773122740992814"/>
        <entailedEvent verb="violate" verbOceanFactor="0.3762997902752764"/>
      </event>
    ▼<event id="200" polarity="POS" synset="tremble#v#1">
        <adverb synset="Still"/>
      </event>
    ▼<event id="201" polarity="POS" synset="take#v#7">
        <subject synset="Eddie Hagen"/>
        <object synset="cigarette#n#1"/>
        <dependent relation="prep:to" synset="lip#n#1"/>
      </event>
    ▼<event id="202" polarity="POS" synset="inhale#v#1">
        <subject synset="Eddie Hagen"/>
        <adverb synset="slowly#r#1"/>
      </event>
    ▼<event id="203" polarity="POS" synset="lie#v#1">
        <adverb synset="back"/>
      </event>
    ▼<event id="204" polarity="POS" synset="think#v#1">
        <subject synset="Eddie Hagen"/>
        <adverb synset="only"/>
        <dependent relation="prep:of" synset="fact#n#1"/>
      </event>
      <event id="205" polarity="POS" synset="on"/>
    </events>
  </context>
```


Figure 24: Part of an extended TE file

Table 15: Time spent to produce extended TE files

Number of files	Average time spent to load TE file (milliseconds /file)	Average time spent on inference (milliseconds/file)	Average time spent to save extended TE file (milliseconds /file))
20	38.10	531.65	68.6

Table 16: Number of entailed event nodes

Number of files	Number of entailed event node
20	959

6.2 Querying the Knowledge

Since the result files are in xml format, XPath[45] and more easily XQuery[46] can be used to answer queries about the facts and entailed events of the stories. Examples of the queries are the following queries:

- List all existing objects in the story in the different contexts.

```
let $doc := .
for $context in $doc//context,
   $object in $context//objects/object
return
   <object synset="{$object/@synset}" context="{$context/@id}">
   </object>
```

Time spent: 82 ms

[45] http://www.w3.org/TR/xpath20/

[46] http://www.w3.org/TR/xquery-30/

- List all states of specific object (e.g. “Eddie Hagen”) expressed in the story in the different contexts.

```
let $doc := .
for $context in $doc//context,
   $state in $context//state,
   $object in $state/modified-object
where
     $ object/@modified-object eq "Eddie Hagen"
return
   <state modifier="{$state/@modifier}" context="{$context/@id}">
   </state>
```

Time spent: 94 ms

- List all actions a specific person (e.g. “Eddie Hagen”) has performed in the story.

```
let $doc := .
for $context in $doc//context,
     $event in $context//event,
     $subject in $event/subject
  where
     $subject/@synset eq "Eddie Hagen"
  return
     (<event event="{$event/@synset}" polarity ="{$event/@polarity}"
context="{$context/@id}">
     </event>, '&#xa;')
```

Time spent: 96 ms

- List all actions a specific person (e.g. “Eddie Hagen”) may have done based on the story. (List all the probable actions which their strength is more than 0.3). Appendix B shows this query and the result.

```
let $doc := .
for $context in $doc//context,
     $event in $context//event,
     $subject in $event/subject,
```

```
    $eevent in $event/entailedEvent
  where
    $subject/@synset eq "Eddie Hagen" and $eevent/@VerbOceanWeight>=0.3
  return
    (<entailedEvent event="{$ eevent /@verb}" polarity ="{$event/@polarity}"
context="{$context/@id}">
    </entailedEvent >, '&#xa;')
```

Time spent: 106 ms

6.3 Generalization of Assertions of Scenes and Contexts

The inference algorithm, and thus the entailment process, focuses on one context at a time. Since all the scenes are sorted based on the associated time, all assertions provided for a particular context are valid in the associated scene, as are further contexts until another evidence or assertion contradicts them. To answer the question of whether the assertions for a context are true in the following contexts, assertions are categorized as: (1) location assertions (2) time assertions (3) event assertions (e.g., “Alex needs the book”) (4) state assertions (e.g., “Alex is short”).To handle generalization over these assertions following strategies are implemented:

- **Location assertions:** locations are saved in the context and scene level in our intermediate structure, but since the concept of location is related to the scenes, the locations are reported in the scene level of text entailment representation. The location in the scene level is provided by generalizing of the locations in the contexts of the current scene. Moving forward from one scene to the next scene, the location of the previous context will not be carried to the next scene, because our scenes cover a part of a story in a specific period of time and location. In the most cases, there may not be explicit cue to change the location in the story, but our system identifies a new scene based on different

features. Therefore, there is a high probability of location change between two scenes even if it is not explicitly expressed in the story.

- **Time assertions:** since time concept is similar to the location concept, our system behaves the same for time and location. Thus, time in the scene level comes from its context level, and time will not be carried by moving forward from one scene to another scene.

- **Event assertions:** an event assertion remains true even if we enter a new context. It only becomes invalid if its contradiction is explicitly expressed in the text. To handle this situation, our system checks all the events in each context with all other events in the following contexts. If there is a pair of events which has the same subject and object, and the two events are in contradiction, a contradicted version of the first assertion will be added to the second context. (e.g. “Alex possess the book” in the second context while we have “Alex needs the book” in the first context causes our system to add an assertion: “Alex doesn’t need the book” to the second context). The contradiction is checked in three different way:

  - Two events are the same but the polarity of them are opposite.
  - Two events are antonyms according to WordNet.
  - Two events have “opposite-of” relation in VerbOcean.

- **State assertions:** a state assertion is similar to an event assertion and it remains true even if we enter to a new context. It only becomes invalid if its contradiction is explicitly expressed afterward. To handle this situation, our system checks all the states in each

context with all other states in the following contexts. If there was a pair of states which has the same modified object and the two states are in contradiction, a contradicted version of the first assertion will be added to the second context. (e.g. “Alex is tall” in the second context while we have “Alex is short” in the first context causes our system to add an assertion: “Alex is not short” to the second context). The contradiction is checked by two different way:

- Two modifiers are the same but the polarity of them are opposite.
- Two modifiers are antonyms according to WordNet.

6.4 Text Entailment Evaluation

Evaluating forward textual entailment is difficult because there is no gold dataset. Hence, for this research, after performing textual entailment process, the assertions are reviewed to determine each assertion is a false assertion or true assertion. Basically, the objective of the evaluation was: How accurate is the system compared to human annotators? The inference is done and entailed events are produced for the contexts in the 20 files. Then, all entailed event were reviewed. Table 17 shows the results of the evaluation.

Table 17: Accuracy of textual entailment

Number of assertions	**Accuracy**
959	73.64%

False instances were investigated to find the error roots. The most decisive source of error was inference rules. VerbOcean as inference rule set is widely used in NLP application and especially in textual entailment, but necessity of more accurate inference rule set incorporating

semantic rules is undeniable. All of VerbOcean's relations are found based on co-occurrence of the verbs on specific patterns in different documents. Figure 25 Shows all relation from verb "accept" and verb "murmur" in VerbOcean file (except for opposite relations). "Accept" is a positive verb. Reader of a text usually after "accept" can entail that a person which accept receive, have, or admit something, but in VerbOcean many negative verbs such as *back*, *cancel*, *return*, and *violate* come after accept. All these events are possible, but the weight assigned to those events are more than its probability in the real world (e.g. weight of relation between *accept* and *return* is almost equal with relation between *accept* and *receive*). The verb "murmur" is paired with verbs which are rather opposite instead of similar.

```
accept [happens-before] back :: 8.759674
accept [happens-before] cancel :: 8.891250
accept [happens-before] defend :: 8.735315
accept [stronger-than] discuss :: 8.962636
accept [happens-before] dismiss :: 9.457765
accept [happens-before] return :: 9.726662
accept [happens-before] spurn :: 12.416029
accept [happens-before] violate :: 9.792037
accept [can-result-in] violate :: 9.579352
accept [stronger-than] discuss :: 8.962636
accept [happens-before] promise :: 9.039170
accept [similar] propose :: 10.369568
accept [similar] consider :: 8.869286
accept [similar] negotiate :: 10.940777
accept [stronger-than] concede :: 11.524982
accept [stronger-than] contemplate :: 10.988839
accept [happens-before] embrace :: 10.762517
accept [happens-before] implement :: 9.338439
accept [similar] receive :: 9.856247
```

```
murmur [happens-before] shout :: 14.789645
murmur [similar] yell :: 11.653064
murmur [similar] chant :: 13.366811
murmur [similar] clap :: 14.006431
murmur [similar] pray :: 9.743947
murmur [similar] roar :: 14.282399
```

Figure 25: Portions of VerbOcean file

VerbOcean also is limited to binary verb entailment rules and neglects semantic (thematic) roles. Therefore, there is no such a rule:

$$\text{If}\ \underbrace{X}_{agent}\ \text{gave}\ \underbrace{Y}_{theme}\ to\ \underbrace{Z}_{recipient} \xRightarrow{entails} \underbrace{Z}_{experiencer}\ have\ \underbrace{Y}_{theme}$$

Such shortcomings of VerbOcean and different applications of inference rule have motivated researchers to start working on providing more beneficial inference rule sets. Weisman and her colleagues (2012) proposed a richer set of linguistic cues for detecting entailment rules between verbs. Mostafazadeh (2015) and Allen are working on learning semantically rich event inference using definition of the verbs. Despite these attempts and ongoing research to provide more beneficial inference rules, no more accurate inference rule is released yet. VerbOcean is still the most accurate available inference rule set to use. However, research systems that use VerbOcean (including ours) inherit its shortcomings.

CHAPTER 7- CONCLUSION AND FUTURE WORK

This research aimed to improve the task of textual entailment, as well as consider the concept of context in the entailment process. To achieve this goal, a context-aware textual entailment system was developed.

A new approach was developed that uses lexical, syntactic, and semantic features to extract the semantic structure of a text. The system identifies scenes and contexts, using a two-phase supervised machine learning method, extended with a voting mechanism.

Moreover, a forward textual entailment system was developed that identifies each context and performs entailment based on that context, as well as background knowledge. This system can be used in many high-level semantic natural language applications, including (but not limited to) question answering, summarization, machine translation, intelligent tutoring systems, word sense and disambiguation.

Important contributions of this work include:

- Developing a two-phase supervised SVM model extended with a voting mechanism for context identification
- Developing an innovative text representation model providing semantic hierarchy
- Developing a textual entailment method using proposed representation of text and entailment rules extracted from background knowledge sources.
- Creating an annotated dataset consisting of context tags for training and testing purposes

7. 1 Recommendations for Future Work

Future work will focus on improving each step of the proposed process:

- **Context identification:** This method can be tuned by exploring new features. Additionally, this research focused on story-like text, the hardest genre for text segmentation methods. Other genres can be investigated.

- **Text representation:** After exploring other major state-of-the-art text representation models, the benefits of these systems can be applied to the proposed text representation model.

- **Gold standard dataset:** Forward textual entailment suffers from the lack of gold standard datasets. Providing the dataset can help textual entailment significantly.

- **Inference Rule set:** Forward textual entailment and many other natural language processing areas also suffers from the lack of highly accurate inference rules. VerbOcean was used in this research, but more accurate inference rule sets incorporating semantics, and providing more than just verb-to-verb entailment would be beneficial.

REFERENCES

1. Baker, Collin F., Charles J. Fillmore, and John B. Lowe. "The berkeley framenet project." In *Proceedings of the 17th International Conference on Computational linguistics*-Volume 1, pp. 86-90. Association for Computational Linguistics, 1998.

2. Bauer, Joseph, Ralf-Detlef Kutsche, and Rudiger Ehrmanntraut. "Identification and modeling of contexts for different information scenarios in air traffic." *Technische Universität Berlin, Diplomarbeit* (2003).

3. Bar-Haim, Roy, Ido Dagan, Iddo Greental, Idan Szpektor, and Moshe Friedman. "Semantic inference at the lexical-syntactic level for textual entailment recognition." In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 131-136. Association for Computational Linguistics, 2007.

4. Bär, Daniel, Chris Biemann, Iryna Gurevych, and Torsten Zesch. "Ukp: Computing semantic textual similarity by combining multiple content similarity measures." In *Proceedings of the Sixth International Workshop on Semantic Evaluation*, pp. 435-440. Association for Computational Linguistics, 2012.

5. Beeferman, Doug, Adam Berger, and John Lafferty. "Statistical models for text segmentation." *Machine learning* 34, no. 1-3 (1999): 177-210.

6. Bentivogli, Luisa, Peter Clark, Ido Dagan, Hoa Dang, and Danilo Giampiccolo. "The seventh pascal recognizing textual entailment challenge."In *Proceedings of Text Analysis Conference (TAC) 2011* (2011).

7. Bjerva, Johannes, Johan Bos, Rob van der Goot, and Malvina Nissim. "The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity." *SemEval 2014* (2014): 642.

8. Bos, Johan, and Katja Markert. "Recognising textual entailment with logical inference." In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 628-635. Association for Computational Linguistics, 2005.

9. Bos, Johan. "Is there a place for logic in recognizing textual entailment?" *Linguistic Issues in Language Technology* 9 (2013).

10. Brockett, Chris. "Aligning the RTE 2006 corpus." *Microsoft Research* (2007).

11. Brown, Peter F., Peter V. Desouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. "Class-based n-gram models of natural language." *Computational linguistics* 18, no. 4 (1992): 467-479.

12. Brown, Peter J. "The stick-e document: a framework for creating context-aware applications." *ELECTRONIC PUBLISHING-CHICHESTER-* 8 (1995): 259-272.

13. Buvač, Saša, Vanja Buvač, and Ian A. Mason. "The semantics of propositional contexts." In *Methodologies for Intelligent Systems*, pp. 468-477. Springer Berlin Heidelberg, 1994.

14. Cai, Deng, Xiaofei He, Kun Zhou, Jiawei Han, and Hujun Bao. "Locality Sensitive Discriminant Analysis." In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 708-713. 2007.

15. Capra, Licia, Wolfgang Emmerich, and Cecilia Mascolo. "Carisma: Context-aware reflective middleware system for mobile applications." *Software Engineering*, IEEE Transactions on 29, no. 10 (2003): 929-945.

16. Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, no. 3 (2011): 27.

17. Chang, Angel X., and Christopher D. Manning. "SUTime: A library for recognizing and normalizing time expressions." In *International Conference on Language Resources and Evaluation*, pp. 3735-3740. 2012.

18. Chklovski, Timothy, and Patrick Pantel. "VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, vol. 2004, pp. 33-40. 2004.

19. Choi, Freddy YY. "Advances in domain independent linear text segmentation." In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pp. 26-33. Association for Computational Linguistics, 2000.

20. Claessen, Koen, and Niklas Sorensson. "New techniques that improve MACE-style finite model finding." In *Proceedings of the CADE-19 Workshop: Model Computation-Principles, Algorithms, Applications*, pp. 11-27. 2003.

21. Clark, Peter, and Phil Harrison. "An inference-based approach to recognizing entailment." In *Proceeding of Text Analysis Conference (TAC)* (2009).

22. Cohen, Jacob. "A coefficient of agreement for nominal scales." *Educational and psychological measurement* 20, no. 1 (1960): 37-46.

23. Cooper, Robin, Dick Crouch, J. V. Eijckl, Chris Fox, J. V. Genabith, J. Japars, Hans Kamp et al. *A framework for computational semantics (FraCaS)*. Technical report, The FraCaS Consortium, 1996.

24. Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. "A framework and graphical development environment for robust NLP tools and applications." In *ACL*, pp. 168-175. 2002.

25. Dagan, Ido, Oren Glickman, and Bernardo Magnini. "The PASCAL recognising textual entailment challenge." In *Machine learning challenges. Evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, pp. 177-190. Springer Berlin Heidelberg, 2006.

26. Dang, Hoa Trang, and Karolina Owczarzak. "Overview of the TAC 2008 update summarization task." In *Proceedings of Text Analysis Conference (TAC)*, pp. 1-16. 2008.

27. De Salvo Braz, Rodrigo, Roxana Girju, Vasin Punyakanok, Dan Roth, and Mark Sammons. "Knowledge representation for semantic entailment and question-answering." In *IJCAI-05 workshop on knowledge and reasoning for answering questions*, pp. 71-80. 2005.

28. De Marneffe, Marie-Catherine, Trond Grenager, Bill MacCartney, Daniel M. Cer, Daniel Ramage, Chloé Kiddon, and Christopher D. Manning. "Robust Graph Alignment Methods for Textual Inference and Machine Reading." In *AAAI Spring Symposium: Machine Reading*, pp. 36-42. 2007.

29. De Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. "Generating typed dependency parses from phrase structure parses." In *Proceedings of International Conference on Language Resources and Evaluation*, vol. 6, no. 2006, pp. 449-454. 2006.

30. De Marneffe, Marie-Catherine, and Christopher D. Manning. "Stanford typed dependencies manual." *URL http://nlp.stanford.edu/software/dependencies manual.pdf* (2008).

31. Del Fabro, Manfred, and Laszlo Böszörmenyi. "State-of-the-art and future challenges in video scene detection: a survey." *Multimedia systems* 19, no. 5 (2013): 427-454.

32. Dethlefs, Nina, and Heriberto Cuayáhuitl. "Hierarchical reinforcement learning for situated natural language generation." *Natural Language Engineering* 21, no. 03 (2015): 391-435.

33. Dey, Anind K. "Understanding and using context." *Personal and ubiquitous computing* 5, no. 1 (2001): 4-7.

34. Du, Lan, Wray L. Buntine, and Mark Johnson. "Topic Segmentation with a Structured Topic Model." In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 190-200. 2013.

35. Dzikovska, Myroslava O., Rodney D. Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa T. Dang. *SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge*. NORTH TEXAS STATE UNIV DENTON, 2013.

36. Fellbaum, Christiane. *WordNet*. Blackwell Publishing Ltd, 1998.

37. Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. "Incorporating non-local information into information extraction systems by gibbs sampling." In *Proceedings of the*

43rd Annual Meeting on Association for Computational Linguistics, pp. 363-370. Association for Computational Linguistics, 2005.

38. Franklin, David, and Joshua Flaschbart. "All gadget and no representation makes jack a dull environment." In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pp. 155-160. 1998.

39. Gabrilovich, Evgeniy, and Shaul Markovitch. "Wikipedia-based semantic interpretation for natural language processing." *Journal of Artificial Intelligence Research* (2009): 443-498.

40. Ghidini, Chiara, and Fausto Giunchiglia. "Local models semantics, or contextual reasoning= locality+ compatibility." *Artificial intelligence* 127, no. 2 (2001): 221-259.

41. Giunchiglia, Fausto. "Contextual reasoning." *Epistemologia, special issue on I Linguaggi e le Macchine* 16 (1993): 345-364.

42. Giunchiglia, Fausto, Vincenzo Maltese, and Biswanath Dutta. "Domains and context: first steps towards managing diversity in knowledge." *Web Semantics: Science, Services and Agents on the World Wide Web* 12 (2012): 53-63.

43. Harabagiu, Sanda, and Andrew Hickl. "Methods for using textual entailment in open-domain question answering." In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 905-912. Association for Computational Linguistics, 2006.

44. Heilman, Michael, and Nitin Madnani. "ETS: domain adaptation and stacking for short answer scoring." In *Proceedings of the 2nd joint conference on lexical and computational semantics*, vol. 2, pp. 275-279. 2013.

45. Heilman, Michael, and Noah A. Smith. "Tree edit models for recognizing textual entailments, paraphrases, and answers to questions." In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1011-1019. Association for Computational Linguistics, 2010.

46. Held, Albert, Sven Buchholz, and Alexander Schill. "Modeling of context information for pervasive computing applications." In *Procceding of the World Multiconference on Systemics, Cybernetics and Informatics*. 2002.

47. Henricksen, Karen, Jadwiga Indulska, and Andry Rakotonirainy. "Generating context management infrastructure from high-level context models." In *In 4th International Conference on Mobile Data Management (MDM)-Industrial Track*. 2003.

48. Hickl, Andrew, and Jeremy Bensley. "A discourse commitment-based framework for recognizing textual entailment." In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 171-176. Association for Computational Linguistics, 2007.

49. Huang, Eric H., Richard Socher, Christopher D. Manning, and Andrew Y. Ng. "Improving word representations via global context and multiple word prototypes." In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 873-882. Association for Computational Linguistics, 2012.

50. Hull, Richard, Philip Neaves, and James Bedford-Roberts. "Towards situated computing." In *Wearable Computers, 1997. Digest of Papers. First International Symposium on*, pp. 146-153. IEEE, 1997.

51. Jiang, Jay J., and David W. Conrath. "Semantic similarity based on corpus statistics and lexical taxonomy." In *Proceedings of International Conference Research on Computational Linguistics* (1997).

52. Katz, Ephraim. "The Film Encyclopedia, revised by Fred Klein and Ronald Dean Nolen." (1998).

53. Klein, Dan, and Christopher D. Manning. "Accurate unlexicalized parsing." In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pp. 423-430. Association for Computational Linguistics, 2003.

54. Knappmeyer, Michael, Saad Liaquat Kiani, Cristina Frà, Boris Moltchanov, and Nigel Baker. "ContextML: a light-weight context representation and context management schema." In *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*, pp. 367-372. IEEE, 2010.

55. Kok, Stanley, Parag Singla, Matthew Richardson, Pedro Domingos, Marc Sumner, Hoifung Poon, and Daniel Lowd. "The Alchemy system for statistical relational AI." *University of Washington, Seattle* (2005).

56. Kotlerman, Lili, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. "Directional distributional similarity for lexical inference." *Natural Language Engineering* 16, no. 04 (2010): 359-389.

57. Kouylekov, Milen, and B. Magnini. "Recognizing Textual Entailment with Tree Edit Distance." In *Proceedings of the PASCAL RTE Challenge*, pp. 17-20. 2005.

58. Kouylekov, Milen, Alessio Bosca, and Luca Dini. "EDITS 3.0 at RTE-7." In *Proceedings of the Seventh Recognizing Textual Entailment Challenge* (2011).

59. Lakoff, George. *Linguistics and natural logic*. Springer Netherlands, 1972.

60. Leacock, Claudia, and Martin Chodorow. "Combining local context and WordNet similarity for word sense identification." *WordNet: An electronic lexical database* 49, no. 2 (1998): 265-283.

61. Lan, Man, Chew Lim Tan, Jian Su, and Yue Lu. "Supervised and traditional term weighting methods for automatic text categorization." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31, no. 4 (2009): 721-735.

62. Lee, Heeyoung, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. "Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task." In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pp. 28-34. Association for Computational Linguistics, 2011.

63. Lee, Heeyoung, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. "Deterministic coreference resolution based on entity-centric, precision-ranked rules." *Computational Linguistics* 39, no. 4 (2013): 885-916.

64. Lin, Dekang. "An information-theoretic definition of similarity." In *ICML*, vol. 98, pp. 296-304. 1998.

65. Lin, Ziheng, Hwee Tou Ng, and Min-Yen Kan. "A PDTB-styled end-to-end discourse parser." *Natural Language Engineering* (2014): 1-34.

66. Lin, Dekang, and Patrick Pantel. "Discovery of inference rules for question-answering." *Natural Language Engineering* 7, no. 04 (2001): 343-360.

67. Lien, Elisabeth. "Using Minimal Recursion Semantics for Entailment Recognition." *EACL 2014* (2014): 76.

68. Llorens, Hector, Estela Saquete, and Borja Navarro. "Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2." In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 284-291. Association for Computational Linguistics, 2010.

69. MacCartney, Bill, Michel Galley, and Christopher D. Manning. "A phrase-based alignment model for natural language inference." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 802-811. Association for Computational Linguistics, 2008.

70. MacCartney, Bill. "Natural language inference." PhD diss., Stanford University, 2009.

71. Malioutov, Igor, and Regina Barzilay. "Minimum cut model for spoken lecture segmentation." In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 25-32. Association for Computational Linguistics, 2006.

72. Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. "The Stanford CoreNLP natural language processing toolkit." In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60. 2014.

73. McCarthy, John. "Notes on formalizing context." (1993).

74. Mehdad, Yashar. "Automatic cost estimation for tree edit distance using particle swarm optimization." In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 289-292. Association for Computational Linguistics, 2009.

75. Mihalcea, Rada, and Dragomir Radev. *Graph-based natural language processing and information retrieval*. Cambridge University Press, 2011.

76. Miller, George, and Christiane Fellbaum. "Wordnet: An electronic lexical database." (1998).

77. Misra, Hemant, François Yvon, Joemon M. Jose, and Olivier Cappe. "Text segmentation via topic modeling: an analytical study." In *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 1553-1556. ACM, 2009.

78. Moro, Andrea, Francesco Cecconi, and Roberto Navigli. "Multilingual Word Sense Disambiguation and Entity Linking for Everybody." Proceedings of the 13th International Conference on Semantic Web (P&D). 2014.

79. Moro, Andrea, Alessandro Raganato, and Roberto Navigli. "Entity linking meets word sense disambiguation: a unified approach." *Transactions of the Association for Computational Linguistics* 2 (2014): 231-244.

80. Mostafazadeh, Nasrin, and James F. Allen. "Learning Semantically Rich Event Inference Rules Using Definition of Verbs." In *Computational Linguistics and Intelligent Text Processing*, pp. 402-416. Springer International Publishing, 2015.

81. Nielsen, Rodney D., Wayne Ward, and James H. Martin. "Recognizing entailment in intelligent tutoring systems." *Natural Language Engineering* 15, no. 04 (2009): 479-501.

82. Padó, Sebastian, Michel Galley, Dan Jurafsky, and Chris Manning. "Robust machine translation evaluation with entailment features." In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pp. 297-305. Association for Computational Linguistics, 2009.

83. Parsons, Terence. *Events in the Semantics of English*. Vol. 5. Cambridge, Ma: MIT Press, 1990.

84. Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. "WordNet:: Similarity: measuring the relatedness of concepts." In *Demonstration papers at Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics 2004*, pp. 38-41. Association for Computational Linguistics, 2004.

85. Perera, Charith, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. "Context aware computing for the internet of things: A survey." *Communications Surveys & Tutorials, IEEE* 16, no. 1 (2014): 414-454.

86. Punyakanok, Vasin, Dan Roth, Wen-tau Yih, and Dav Zimak. "Semantic role labeling via integer linear programming inference." In *Proceedings of the 20th International Conference on Computational Linguistics*, p. 1346. Association for Computational Linguistics, 2004.

87. Pustejovsky, James, José M. Castano, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. "TimeML: Robust Specification of Event and Temporal Expressions in Text." *New directions in Question Answering* 3 (2003): 28-34.

88. Resnik, Philip. "Using information content to evaluate semantic similarity in a taxonomy." In *Proceedings of the 1995 International Joint Conference on AI (IJCAI-95)* (1995).

89. Riazanov, Alexandre, and Andrei Voronkov. "The design and implementation of VAMPIRE." *AI Communications* 15, no. 2 (2002): 91-110.

90. Richardson, Matthew, and Pedro Domingos. "Markov logic networks." *Machine Learning* 62, no. 1-2 (2006): 107-136.

91. Roth, Dan, Mark Sammons, and V. G. Vydiswaran. "A framework for entailed relation recognition." In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 57-60. Association for Computational Linguistics, 2009.

92. Rudzewitz, Björn, and Ramon Ziai. "CoMiC: Adapting a Short Answer Assessment System for Answer Selection." In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval*, vol. 15. 2015.

93. Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." *Communications of the ACM* 18, no. 11 (1975): 613-620.

94. Schilit, Bill, Norman Adams, and Roy Want. "Context-aware computing applications." In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pp. 85-90. IEEE, 1994.

95. Schmitz, Michael, Robert Bart, Stephen Soderland, and Oren Etzioni. "Open language learning for information extraction." In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 523-534. Association for Computational Linguistics, 2012.

96. Schubert, Lenhart K., and Chung Hee Hwang. "Episodic Logic meets Little Red Riding Hood: A comprehensive, natural representation for language understanding." *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language* (2000): 111-174.

97. Schubert, Lenhart K., Benjamin David Van Durme, and Marzieh Bazrafshan. "Entailment Inference in a Natural Logic-like General Reasoner." In *AAAI Fall Symposium: Commonsense Knowledge*. 2010.

98. Schuler, Karin Kipper. "VerbNet: A broad-coverage, comprehensive verb lexicon." (2005).

99. Shnarch, Eyal, Libby Barak, and Ido Dagan. "Extracting lexical reference rules from Wikipedia." In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pp. 450-458. Association for Computational Linguistics, 2009.

100. Socher, Richard, John Bauer, Christopher D. Manning, and Andrew Y. Ng. "Parsing with compositional vector grammars." In *In Proceedings of the Association for Computational Linguistics Conference*. 2013.

101. Socher, Richard, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. "Grounded compositional semantics for finding and describing images with sentences." *Transactions of the Association for Computational Linguistics* 2 (2014): 207-218.

102. Strang, Thomas, Claudia Linnhoff-Popien, and Korbinian Frank. "CoOL: A context ontology language to enable contextual interoperability." In *Distributed applications and interoperable systems*, pp. 236-247. Springer Berlin Heidelberg, 2003.

103. Strang, Thomas, and Claudia Linnhoff-Popien. "A context modeling survey." In *Proceedings First International on Advanced Context Modeling, Reasoning, and Management*. 2004.

104. Stern, Asher, Amnon Lotan, Shachar Mirkin, Eyal Shnarch, Lili Kotlerman, Jonathan Berant, and Ido Dagan. "Knowledge and tree-edits in learnable entailment proofs." (2011).

105. Tan, Yongmei, Junyu Zeng, Xiaojie Wang, and Eduard Hovy. "BUPTTeam participation at TAC 2011 recognizing textual entailment." *Proceedings of Text Analysis Conference* (2011).

106. Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. "Feature-rich part-of-speech tagging with a cyclic dependency network." In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pp. 173-180. Association for Computational Linguistics, 2003.

107. Toutanova, Kristina, and Christopher D. Manning. "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger." In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pp. 63-70. Association for Computational Linguistics, 2000.

108. Tsuchida, Masaaki, and Kai Ishikawa. "IKOMA at TAC2011: A Method for Recognizing Textual Entailment using Lexical-level and Sentence Structure-level features." *Proceeding of Text Analysis Conference* (2011).

109. UzZaman, Naushad, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. "Tempeval-3: Evaluating events, time expressions, and temporal relations. In *Proceedings of the International Workshop on Semantic Evaluation* (2012).

110. Van Benthem, Johan. "The lambek calculus." In *Categorial Grammars and Natural Language Structures*, pp. 35-68. Springer Netherlands, 1988.

111. Wang, Xiao Hang, Da Qing Zhang, Tao Gu, and Hung Keng Pung. "Ontology based context modeling and reasoning using OWL." In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pp. 18-22. IEEE, 2004.

112. Wang, Jinqiao, Lingyu Duan, Qingshan Liu, Hanqing Lu, and Jesse S. Jin. "A multimodal scheme for program segmentation and representation in broadcast video streams." *Multimedia, IEEE Transactions on* 10, no. 3 (2008): 393-408.

113. Wei, Chih-Ping, Christopher C. Yang, and Chia-Min Lin. "A Latent Semantic Indexing-based approach to multilingual document clustering." *Decision Support Systems* 45, no. 3 (2008): 606-620.

114. Weisman, Hila, Jonathan Berant, Idan Szpektor, and Ido Dagan. "Learning verb inference rules from linguistically-motivated evidence." In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 194-204. Association for Computational Linguistics, 2012.

115. Wu, Zhibiao, and Martha Palmer. "Verbs semantics and lexical selection." In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, pp. 133-138. Association for Computational Linguistics, 1994.

116. Yokote, Ken-ichi, Shohei Tanaka, and Mitsuru Ishizuka. "Effects of using simple semantic similarity on textual entailment recognition." Proceedings of *Text Analysis Conference* (2011).

117. Zanzotto, Fabio Massimo, Lorenzo Dell'Arciprete, and Alessandro Moschitti. "Efficient Graph Kernels for Textual Entailment Recognition." *Fundamenta Informaticae* 107, no. 2 (2011): 199-222.

118. Zhai, Yun, Alper Yilmaz, and Mubarak Shah. "Story segmentation in news videos using visual and text cues." In *Image and Video Retrieval*, pp. 92-102. Springer Berlin Heidelberg, 2005.

VITA

Soha Khazaeli received her bachelor of science at Shahid Beheshti University in Computer Engineering in 2001. Thereafter, she studied Masters and received her degree in Computer Engineering (Software) at Isfahan University in 2004. Then she worked as a faculty in Electrical and Computer Department of Babol University of Technology for six years. She started graduate studies in the college of engineering at Louisiana State University (LSU) as a teaching assistant in Engineering Science department. During her years at LSU she has worked as an instructor and teacher assistant.

She is a candidate for the Doctor of Philosophy degree in Engineering Science with concentration in Information Technology and Engineering (ITE) under supervision of Dr. Gerald Knapp. The degree will be conferred at the Fall commencement 2015.