AN UNSUPERVISED LEARNING APPROACH FOR AUTOMATIC DETECTION OF METAPHORS

By

DANIEL A. NIETEN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2011

To Teresa and Brandon

ACKNOWLEDGMENTS

I would like to thank my wife Teresa and son Brandon for their love, support, patience, and understanding for the time this work took away from them. Christine and Dave Langhorne for proof reading my early drafts. My cousin Sean Smith for reading numerous electronic books and creating the gold data set used in the final analysis. My brother Joe Nieten for his continuous encouragement and for suffering through the laborious effort associated with proof reading the manuscript. Finally, I want to thank my entire family, without their continued support I would never have been able to achieve this dream.

I would like to thank my advisor, Professor Paul Fishwick, whose guidance, encouragement, support, and patience have allowed me to complete this work. I would also like to thank the members of my committee; Professors Douglas Dankel, Abdelsalam (Sumi) Helal, Gary Koehler, and Joe Wilson for their time and effort. I thank Mr. John Bowers for all his support and encouragement over the years. I would like to thank Dr. Kent Bimson and Dr. Richard Hull, to who I owe a great deal of thanks for their interest in this work and use of their expertise in the field of linguistics.

This work was sponsored in part by the Office of Naval Research (ONR), Navy SBIR 2010.2 - Topic N102-176.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACM	Association for Computing Machines
AI	Artificial Intelligence
ANNIE	A Nearly-New Information Extraction System
ATT-Meta	Attribute Metaphor
CCS	Computing Classification System
CMI	Computational Metaphor Identification
CS	Collative Semantics
EM	Expectation Maximization
FST	Finite State Transducer
GATE	General Architecture for Text Engineering
IARPA	Advanced Research Projects Activity
JAPE	Java Annotation Pattern Engine
KL-ONE	Knowledge representation system
KODIAK	Knowledge representation language
kNN	k Nearest Neighbor
MES	Metaphor Extended System
MIDAS	Metaphor Interpretation, Denotation and Acquisition System
MIS	Metaphor Interpretation System
MET*	Metaphor Star
MIP	Metaphor Identification Procedure
MP	Mapping Principle
NLP	Natural Language Processing
NLTK	Natural Language Tool Kit
OWL	Web Ontology Language

PCFG	Probabilistic Context Free Grammars
POS	Part of Speech
PS	Preference Semantics
RDF	Resource Description Framework
RFI	Request for Information
SPARQL	SPARQL Protocol and RDF Query Language
SVM	Support Vector Machine
TroFi	Trope Finder
UC	Unix Consultant
W3C	World Wide Web Consortium
WSD	Word Sense Disambiguation
WSJ	Wall Street Journal

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

AN UNSUPERVISED LEARNING APPROACH FOR AUTOMATIC DETECTION OF METAPHORS

By

Daniel A. Nieten

August 2011

Chair: Paul Fishwick
Major: Computer Engineering

Metaphors were once thought to only be rhetorical and decorative language relegated to poets and politicians with no cognitive significance. In reality metaphors are pervasive in verbal, written, and visual communications and are key in the methods that we use to learn and recall information. They are used to describe the similarities between two different concepts, to provide a symbolic representation of something abstract, to define relationships, interactions, or mappings between two concepts, object, events, or ideas that are literally not related and not just as rhetorical devices. This makes metaphors essential as teaching devices and of significant value in communication and not just nice decorative language. However, due to the abstract nature of the concepts that compose metaphors, they can be difficult for Natural Language Processing (NLP) tools to detect and categorize.

The importance of detecting figures of speech such as metaphors using NLP tools is simple, people do not communicate in purely concrete language, they employ abstract language such as metaphors, analogy, etc. The processing of documents is laborious at best making detection an ideal task for a computer. However, if the NLP tools are only capable of processing concrete language then many aspects of what is

communicated will be missed or misinterpreted as illogical. If the NLP tools can detect metaphors in written communication then the concrete language content can be used to establish additional context for identifying and classifying the metaphor.

Here we present a method that leverages syntactic part-of-speech tagging in conjunction with semantic relationships defined in lexical knowledge bases to detect linguistic metaphor candidates. The hybrid algorithm calculates a confidence factor for each sentence indicating if there are metaphorical characteristics. The experimental results demonstrate the methodology can contribute to the automated detection of metaphorical candidates.

CHAPTER 1
INTRODUCTION AND MOTIVATIONS

> The greatest thing by far is to have a command of metaphor. This alone cannot be imparted by another; it is the mark of genius, for to make good metaphors implies an eye for resemblances.
>
> —Aristotle, *Poetics* [1]

What Are Metaphors

Within literature (e.g., books, poems, magazines, comic books), writers use words both literally and figuratively. The linguistic constructs that are used to express concepts figuratively are called figures of speech and are a mechanism writers can utilize to make words more colorful and interesting. The differences between these figures of speech can be subtle but very important. Metaphors can be described as X is Y or in mathematical terms as X = Y. There are a number of figures of speech closely related to metaphor include; analogy, tropes, idioms, and simile.

Idioms

An idiom is a phrase that has an overall different meaning from the meaning of its individual parts. Take the example; "to burn the candle at both ends" means to exhaust one’s self by attempting to do too much. Burning a candle at both ends makes very little sense, if a candle is lit at both ends, it would burn out twice as fast. Looking at the overall meaning if a person is doing too much, they too could burn out quickly. The more abstract an idiom the more difficult it is to determine the intended meaning. Table 1-1 shows several common English idioms and their intended meanings.

Table 1-1. Example idioms and the intended meanings

Idiom	Meaning
he was born with a silver spoon in his mouth	he was born rich
it makes my mouth water	it makes me hungry
it was music to my ears	I received good news
let's play it by ear	let's improvise

Trope

In linguistics a trope is a rhetorical figure of speech that uses a word in a way other than what is traditionally considered the literal or normal form providing a 'play on words'. Using a trope is a way for an author to turn a word into something other than the normal or accepted meaning. The term trope is derived from the ancient Greek word τρόπος – tropos, to turn or change direction. There are six common types of tropes; irony, allegory, synecdoche, antanaclasis, metonymy, and metaphor. Metaphor is the only trope within the scope of this research.

Table 1-2. Examples of tropes

Tropes	Definition	Example
Irony	A statement in which the meaning employed is different from the meaning expressed	As *clear* as mud[1]
Allegory	Conveying a story meaning apart from the literal meaning	The Ant and the Grasshopper
Synecdoche	Figure of speech where the part is used to represent the whole	All *hands* on deck[2]
Antanaclasis[3]	A word used in two or more of its possible meanings (i.e., sleep is to rest, sleep is death).	The woods are lovely, dark, and deep, But I have promises to keep, And miles to go before I sleep, And miles to go before I sleep.
Metonymy	Where a thing or concept is not referred to by its original name, but the name of something immediately associated with the thing or concept.	Lend me your *ear*[4]
Metaphor	Where a word or phrase is used to describe an object to which it is not literally related.	John *gave* Mary a cold

[1] http://en.wikipedia.org/wiki/Irony

[2] http://en.wikipedia.org/wiki/Synecdoche

[3] antanaclasis. (2011). In *Encyclopædia Britannica*. Retrieved from http://www.britannica.com/EBchecked/topic/26958/antanaclasis

[4] http://en.wikipedia.org/wiki/Metonymy

Analogy

An analogy is a comparison between two different words or things used to emphasize a point of similarity. Analogies are expressed in the form X is to Y. It should be noted that while analogies and metaphors share certain similarities they are not the same. Analogy is primarily a figure of language that expresses a set of like relationships between two sets of terms. Unlike metaphor, an analogy only claims similarity and does not claim complete identification of the relationships. Table 1-3 provides a small set of analogy examples.

Table 1-3. Example analogies and the intended meanings

Analogies	Meaning
the relationship between them began to thaw	This means that the personal feelings are diminishing
her laugh is as annoying as nails on a chalkboard	This means that her laugh is difficult to listen to
I am going to be toast when I get home	This means that the speaker is in trouble when they arrive home
he is like a rock	This means he is steadfast, strong, and reliable
I feel like a fish out of water	This means the speaker is not comfortable in their current surroundings
there are plenty of fish in the sea	This means that a person is not the only one, there are others

Simile

A simile is a figure of speech comparing two unlike things, often introduced with the word "like" or "as". In many aspects simile is just a specific type of metaphor since they are both forms of comparison, similes allow the two ideas to remain distinct in spite of their similarities, whereas metaphors compare two things without using "like" or "as". The intent is for the reader to picture an item as being like another item, in a metaphor the intent is to portray one item as actually being another (see Table 1-4).

Table 1-4. Examples of similes and metaphors

Simile	Metaphor
her hair was like silk	her hair was silk
mean as Oscar the Grouch	meaner than Oscar the Grouch
the ship went down like lead	dead fish are polished marble
light as a feather	those figures are fishy
busy as a bee	car salesmen are sharks
her gaze was like ice	her gaze was icy

Metaphor

The term metaphor derives from the Greek word μεταφορά (metaphora) meaning "transference" and from the Greek word μεταφέρω (metaphero) or "to carry over, to transfer". In literature, metaphor is a figure of speech or phrase where one word or is being equated in some way to another word. Richards defines a metaphor as having two parts - the tenor and vehicle [91]. The vehicle is the SOURCE from where attributes are derived and tenor is the TARGET of the conceptual attributes. A traditional metaphor is structured in the form: X is (like) Y, where X is the SOURCE domain concept and Y is the TARGET domain concept (Ex: “Life is a journey”). The SOURCE domain of a metaphor is typically concrete and related to something that is well understood (e.g., journey). The TARGET domain is the aspect that is very abstract or not well understood typically based on a psychological state, personal, social or cultural experiences or traditions (e.g., Life). Table 1-5 describes different metaphor types and the corresponding SOURCE and TARGET domain associations. The relationship between the SOURCE and TARGET domains is defined by mapping principles that describe the analogical reasoning or inference that describes/defines the metaphor. Mathematically metaphors are represented as X = Y, grammatically they can be represented as a comparison of two subjects without using 'like' or 'as' or as X is Y. This device is well known for usage in poetry, where with few words, emotions and

associations from one context are associated with objects and entities in a different context.

Table 1-5. Types of metaphors with SOURCE and TARGET domain associations

Metaphor Type	Description
Absolute (paralogical or antimetaphor)	Nonsensical, there is no known relationship between the SOURCE and TARGET domains. Example: " We are at the dog end of every task."
Active or new	Relatively new, has not become part of everyday linguistic usage. The audience may or may not know that a metaphor has been used, requires explanation. Example: "You are my sun."
Complex, linguistic, or conventional	Presents the TARGET domain in the terms of the SOURCE domain. Example: "That sheds some light on the subject." Shedding light is a metaphor and there actually is no light.
Compound	Combines several conceptual points with salient or similar aspects can us adjectives and adverbs. Example: "He has the wild stag's foot." This phrase suggests grace and speed as well as daring.[5]
Dead	Where the specific metaphor has been used so much the SOURCE and TARGET domains are not different concepts. Example: “To run for office.”
Dormant	Where the what, who, or why aspect of the SOURCE or TARGET domains are missing (i.e., the sentence has been truncated). Example: “He was caught in the heat of the moment.”
Extended or telescoping	Expands the primary TARGET concept by using several subsidiary concepts. Example: “All the world’s a stage and humans are merely players.”
Implicit (Implied), Submerged	Either the SOURCE or TARGET domain is implied instead of being explicitly defined. Example: “She stroked his fur the wrong way.”
Mixed	Is a combination of two or more SOURCE and TARGET domains that are not directly related. Example: “We all know the leopard can't change his stripes.” Al Gore (The Toronto Sun, 11/19/95)
Root or conceptual	Generalizations, other metaphors can be derived from them. Examples “Conflict is war.”; “Time is Money.”
Simple	There is a single point of resemblance between the SOURCE and TARGET domains. Example: “Chill out!”
Submerged	The SOURCE domain is implied, or indicated by a single aspect. Example: “He hoofed it.”

5 http://www.spiritus-temporis.com/metaphor/types-of-metaphor.html

Why Metaphors Matter

Humans do not communicate or learn in a purely concrete manner. We use symbolism and other abstract forms to communicate ideas and create associations between concepts we understand and those we find difficult to express. Metaphors are one form of symbolism once thought to only be rhetorical and decorative language relegated to poets and politicians with no cognitive significance. In reality metaphors are pervasive in verbal, written, and visual communications and are key in the methods that we use to learn and recall information permeating human thought and action.

Metaphors are of critical interest in the areas of psychology [45] [43], psycholinguists [79][80], psychotherapists [63], cognitive scientists [60], and cognitive linguists [6]. In general the research has shown metaphors are used to describe the similarities between different concepts, to provide a symbolic representation of something abstract, to define relationships, interactions, or mappings between two concepts, objects, events, or ideas that are literally not related and not just as rhetorical devices. This makes metaphors essential as communication and teaching devices and therefore of significant value in communication and knowledge retention [8] and not just nice decorative language.

Lakoff and Johnson [60] discuss metaphors from a cognitive perspective where metaphors are not only used to describe concepts but are used to present facts, to win arguments, and to influence the beliefs and opinions of others. Kovecses has created an extensive introduction to metaphors [56] intended to provide a collection of information covering recent activity. The following list is not intended to be comprehensive but to provide a general idea of the diverse breadth of research interests associated with metaphors.

- The neural theory of metaphor
- The theory of conceptual metaphor
- Metaphor in discourse
- The relationship between embodiment and metaphor
- The embeddings of metaphor in cultural context
- The nature of mappings
- Metaphor of gestures
- The study of multimodal metaphor
- Metaphor identification (detection)
- Metaphor processing
- The corpus linguistics study of metaphor
- Emotion metaphors
- The theory of metonymy
- Metaphors in foreign language teaching
- Metaphors in the study of grammar
- Metaphors in teaching and concept retention

This indicates that metaphor research is currently a very open and active area crossing multiple disciplines including psychology, psycholinguistics, cognitive science and linguistics. Comprehensive coverage of this material is beyond the scope of this research, however, these areas are presented in detail within the individual works of Gentner [42][43], Gibbs [45][46][47], Weiner [103], Ortnoy [79][80], Hayakawa [50], Richards [91], Lawley [63], among many others. They are referenced here because they present many of the motivations for utilizing computers to automatically detect and classify metaphors within specific corpora including those composed of classic published literature, web blogs, emails, news stories, political speeches, and religious sermons. This research focuses on the specific area of metaphors in language and the theoretical and practical application of semantics, or the meaning of the words within a context, and how semantics can be applied to the unsupervised detection and classification of metaphors.

Why Study Metaphors

On November 19, 1863 Abraham Lincoln delivered his famous Gettysburg address. The entire speech was only 243 words in length but is one extended metaphor describing the United States and individual Americans in terms of conception, birth, fighting, and death. Similar uses of metaphors have been employed throughout history to stimulate emotions and motivate individuals, social and political groups, and even entire cultures. Metaphors affect the way we see, think, act, argue, learn, and communicate. Metaphors can also constrain, limit, and actually create a negative reality [50][91]. Kenneth Burke [14] calls this affect “terministic screen”, where a set of symbols provides a perspective or canvas of intelligibility through which the world makes sense to each of us individually. Burke also offers a mechanism for understanding the relationship between language and ideology. Burke viewed language as something that does not simply reflect reality, but also helps deflect reality.

> Metaphors may create realities for us, especially social realities. A metaphor may thus be a guide for future action. Such actions will, of course, fit the metaphor. This will, in turn, reinforce the power of the metaphor to make experience coherent. In this sense metaphors can be self-fulfilling prophecies.
>
> —Lakoff [61]

Unfortunately this theory has proven true throughout history where metaphorical language has been used to sway the opinions and perspective of an individual or entire populace against another religious or political faction within a country [50] or a neighboring country. Even the inaugural speech of President Obama contained numerous metaphors comparing prosperity with tidal flows, political trouble with severe meteorological events, and peace in terms of placid waters. Statistically humans utter about one metaphor every six minutes or every ten to twenty five words [41].

Recognition that metaphors are pervasive and more than decorative language means that in addition to the typical approach taken by computational linguistics methods that employ syntactic grammar to parse sentences [20][21], perform lexical parsing [25][28], and tagging [26] associated with natural languages. The methodology and processing also need to include semantic processing to determine the meaning within a context of the corpus and the culture where text originated. It should be noted that while Lakoff and Johnson [61] present the initial theory associating metaphoric symbolism and cognitive thought process, they do not discuss how metaphors are computationally identified within the text, only that the metaphors can be classified within the metaphor and concept hierarchy described in the Master Metaphor list [62].

This makes the automatic detection, classification, and mapping of metaphors of particular interest to psychotherapist, advertising agencies, clinical researchers, law enforcement agencies, and intelligence analysts as well as the traditional literature researcher, while presenting a significant issue for any individual attempting to explain cognitive thought, or anyone trying to train a computer to process and understand natural language.

A linguistic or conventional metaphor is used to provide a level of symbolism for expressing underlying concepts, emotions, and ideas that can be an individual perspective or a concept shared within a specific culture. However, linguistic metaphors do not typically follow the traditional syntactic structure identified with traditional metaphors (Ex: “Time is running out”; “She was at a crossroads”). Both traditional and linguistic metaphors are organized associations of abstract concepts in the TARGET

domain in terms of concrete concepts in the SOURCE domain defined in cognitive metaphor theory and the master metaphor list for conceptual metaphors [60][62].

Gibbs challenges the 'literal meaning hypothesis' [45] pointing out that traditional theories assert that every sentence has a literal meaning. This hypothesis is obviously false, no one would take the statement "Sarah 'Barracuda' Palin and the Piranhas of the Press"[6] literally. Sarah Palin is certainly not an aggressive saltwater fish of the genus Sphyraena[7], and while it might be subject to debate, the press is not a school of piranhas. The hypothesis expects that the listener or reader will use background assumptions or other a priori knowledge along with contextual information to comprehend the sentence. In later research Gibbs [46] looked at how context might influence the cognitive effects on understanding of a particular metaphor for example.

"Lawyers are sharks". This statement is reasonably straightforward for a human to understand regardless of the context and the literal translation. There is an implication in both of the previous examples, where lawyers, the press, and Sarah Palin may each exhibit behavior that is similar to those observed for sharks, piranhas, and barracuda respectively.

Problem Statement

Metaphors can be described as X is (like) Y or in mathematical terms as X = Y. While this seems simple and straight forward, X and Y are actually very difficult to relate linguistically. Syntactically metaphors are no different than other English sentences and phrases; however, when they are examined semantically at the level of the meaning for

[6] http://www.politicsdaily.com/2009/07/08/sarah-barracuda-palin-and-the-piranhas-of-the-press/

[7] http://www.webster-dictionary.org/definition/genus%20Sphyraena

a specific word within the context of the individual sentence or within the context of the specific document or an entire corpus, another set of rules is observed and required for the identification of a candidate metaphor. There are a number of grammatical and semantic linguistic features that must be considered when attempting to computationally identify metaphors. The actual part of speech; is the word a noun, verb, adjective, preposition, or adverb, is the word plural, what is the word sense. Depending on the context of a given document a sentence could be intended to be literal or metaphorical in nature. The X = Y relationship implies that X and Y are literally related in some manner. Unfortunately, the 'Is-A' relationship is not binary and as such requires determining if X is a synonym of Y, where there could be multiple synonyms for each word represented by X and Y. To further complicate the evaluation linguistically X and Y can satisfy the X=Y relationship using the 'Part-Of' relationship or hyponym attribute. Identifying the meaning of the words in a given sentence within a social of cultural context could be the difference between understanding something that is culturally, socially, or individually significant or misinterpreting the meaning resulting in an unnecessary misunderstanding or conflict.

The detection of a metaphor is dependent on the ability to identify or learn the context of not only the words in each sentence in order to determine the parts of speech on the grammatically syntactic level but also the context of the text that contains the sentence of interest. The following statements are examples where depending on the larger context of utterance that could be construed as a literal or only as a metaphoric description of an event.

- "The defense killed the quarterback."
- "The defense sacked the quarterback."

Both statements appearing in the context of an American sports story invoke the image of a defensive team running toward and tackling the other team's quarterback in a rather harsh fashion. The majority of Americans would certainly comprehend both statements as the meanings of the words are not culturally opaque. However, in the context of a police or accident report, they would be assumed as having the literal meaning where the quarterback was actually killed or robbed (sacked) by one or more of the defensive team members.

Solution

Previous work on the difficult problem of metaphor detection including Martin's MIDAS [70] and Fass' met* [37] are based on a knowledge model where the natural language system identifies metaphor candidates to specialized metaphor handling components. These components analyze the candidate based on a set of hand-coded rules used to determine if the utterance is literal or metaphorical in nature. Each was specifically focused on two semantic domains (e.g., plants and people) and interpreting the metaphor candidate based on linkages to the pre-defined set of conceptual metaphorical domains and for violating some semantic rule. An example of how a rule functions in Martin's system; when presented with the sentence "Robbie's running ate up the distance", the system links the verb-object pair defined in the Eat-Up/Reduce-Money. This is based on information in the knowledge base that distance and money are analogous because they can both be reduced in amount and therefore the program substitutes distance for money. These hand coded semantic rules present a significant limitation to the general detection of metaphorical language. Mason's CorMet [73] is a corpus based method for semi-automatically finding metaphoric mappings between

concepts. CorMet uses a common selectional preference gradient between domains and can within a limited scope identify metaphoric language. The primary improvement over other computational approaches to metaphor detection is that it does not require a manually created knowledge base other than WordNet. WordNet is a machine-readable dictionary containing lexical and semantic information. CorMet also has a serious limitation in that it can only identify metaphoric language if it manifests itself as a common selectional preference (e.g., noun – verb pair).

Each of these solutions share similar problems; they are very limited in scope because of the restricted domain and the limited number of hand constructed rules, as Dolan so eloquently stated all these solutions can not be considered more than a “toy” system [30]. Increasing the scope to support a large corpus of free text would involve more than just increasing the number of rules. The systems also needs to account for word senses and a number of other semantic relationships.

The purpose of this research was to create a methodology, realized in a software tool, which considers the grammatical and semantic relationships along with machine-readable dictionary containing both lexical and semantic information to perform the detection of metaphor candidates given a corpus of documents. This method employs a hybridized approach of statistical and artificial intelligence techniques, using an unsupervised clustering progressive model to quickly bootstrap the semantic information for each sentence and determine the most desirable metaphor candidates. The resulting tool can be used during corpus ingest to identify candidate metaphors and the salient characteristics that support the assertion that the given sentence is a

metaphor. It can be used as a preprocessing step to create a list of candidate linguistic metaphors that can be mapped to conceptual metaphors.

Contributions to Knowledge

The contributions from this research are in two areas – 1) the use of heuristics based on the X is (like) Y, statistical methods, unsupervised learning/clustering techniques to identify salient characteristics between the individual words in a given sentence utilizing machine-readable dictionaries containing lexical and semantic information, and 2) the use of a confidence factor to indicate metaphor candidate since detection is not a binary (true/false). The goal of contribution # 1 is to demonstrate that there is sufficient grammatical/lexical and semantic information that can be assembled in an unsupervised manner to bootstrap the detection of metaphors using existing machine-readable dictionaries and other lexical knowledge basis (e.g., WordNet) thus reducing or removing the need to create rules by hand for even a small domain. The goal of contribution # 2 is to delineate metadata associated with a given sentence to calculate a confidence factor based on the lexical semantic relations between the SOURCE and TARGET domains.

Research Content Outline

This research presents a methodology for automatically detecting metaphor candidates contained in a given text. We investigated why metaphor detection is hard, who is interested in metaphor detection, related work, our algorithm, analysis of experimentation results, and conclusions. The remainder of this research proceeds as follows:

- Chapter 2: Metaphors Detection: This chapter presents several Use Case that help to define the areas where metaphors are being researched, the problem

space associated with metaphors, and the significant challenges associated with detecting metaphorical utterances in a given text.

- Chapter 3: Background and Related Work: This chapter presents background and detailed information on related research efforts associated with the detection and interpretation of linguistic metaphors, the mission pieces for metaphor detection, and the contribution made through this research to computer science and computational linguistics.

- Chapter 4: Prototype: This chapter presents the prototype(s) developed to demonstrate the research and the experiment scenarios.

- Chapter 5: Analysis and Results: This chapter presents the analysis on the experimental results generated from the prototype(s) along with the results from the experiment scenarios.

- Chapter 6: Conclusions and Future Work: This chapter presents the conclusions for this research including the variations in the prototype, what was learned from this research, and what future work could be done to extend or improve this research.

CHAPTER 2
METAPHOR DETECTION

Use Cases

In general the area of metaphor research is extremely diverse, including areas of cognitive psychology, psycholinguistics, psychotherapy, cognitive science, and cognitive linguistics to name a few. The overall diversity of metaphor research has resulted in a number of theories definitions, descriptions, hypotheses, opinions, research, and general explanations given by researchers across multiple disciplines emphasizing the importance of metaphor phenomenon. Bowdle and Gentner [13] present a hypothesis that postulates a concept mapping of metaphors from comparison to categorization, later they also presented an approach to unifying metaphor with analogy and similarity [43]. Burke [14] was focused on the symbolic actions associated with metaphors. Carroll and Mack [16] performed analysis of metaphor in an attempt to explain why metaphors are intrinsically open-ended and how their open-endedness stimulates the construction of mental models. Gentner and Grudin [42] investigated how metaphors change over time and how this affects models of the human mind.

Gibbs [45] assessed the relationship between literal meaning and the psychological understanding of natural language and how this affects the understanding of non-literal discourse. Gibbs continued research into the cognitive effort required to comprehend metaphorical utterances as opposed to literal speech in the area of psycholinguistics [46][47]. Ortony [79][80] also researched the psycholinguistic aspects of metaphors and that metaphors are essential to communication and of great educational value. Metaphors are studied and critiqued to help find out how both the author thinks and how the communication may entice or convince others to think and

the implications of how they can influence the political, social, and cultural perspectives of people sharing the metaphor concepts.

Intelligence Analyst

In September 2009 the Intelligence Advanced Research Projects Activity (IARPA) published a Request for Information (RFI) [8] seeking information on the challenges associated with the theory and current state of the technologies related to the detection and analysis of metaphor usage. The IARPA interest is based on the premise that analysis of how metaphors are used by people in oral and written communication can provide valuable insights into the similarities and differences in worldview across cultures and subcultures.

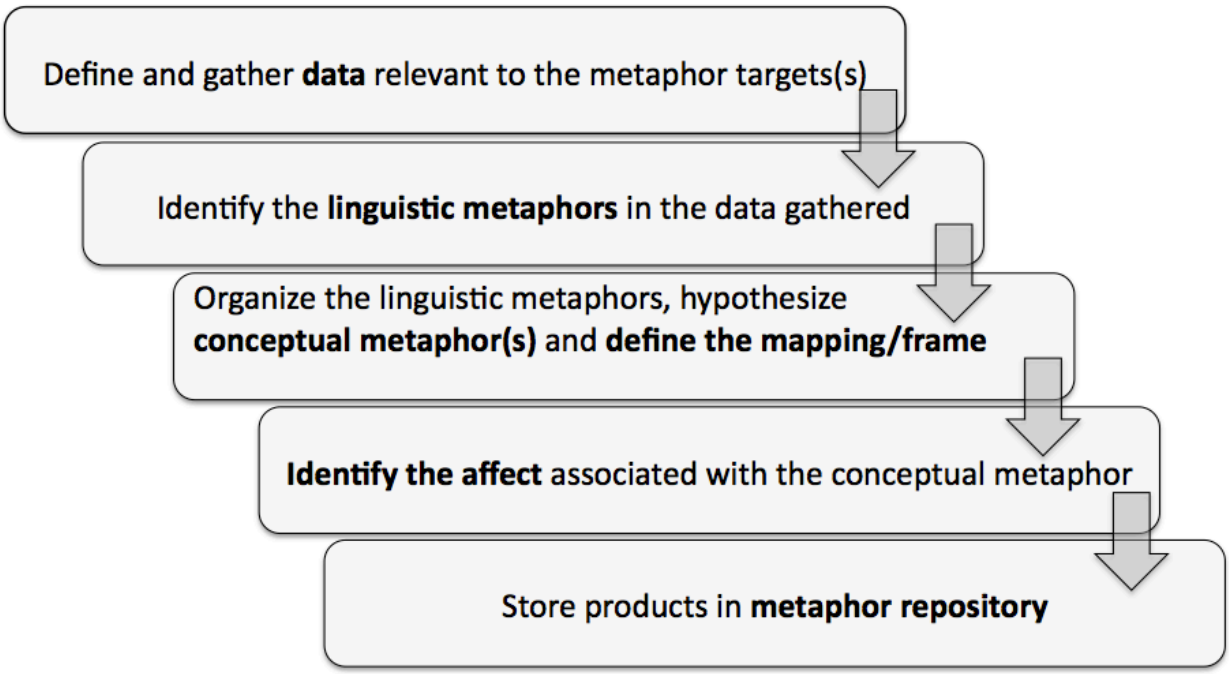

Figure 2-1. Intelligence analyst metaphor processing steps (credit: IARPA)

[8] http://www.iarpa.gov/rfi_META.html

An analyst needs to be aware of worldviews and the various cross-cultural problems that can be encountered. However, the identification and understanding of culturally patterned behavior and shared concepts tend to be opaque even to natives of the cultural, social, or political group. The ability to detect metaphors defining cultural contrasts used for abstract and social concepts: Life is a Game vs. Life is a Struggle. Allowing an analyst to view a structured representation of the metaphors potentially exposing a protagonist's hidden views and goals.

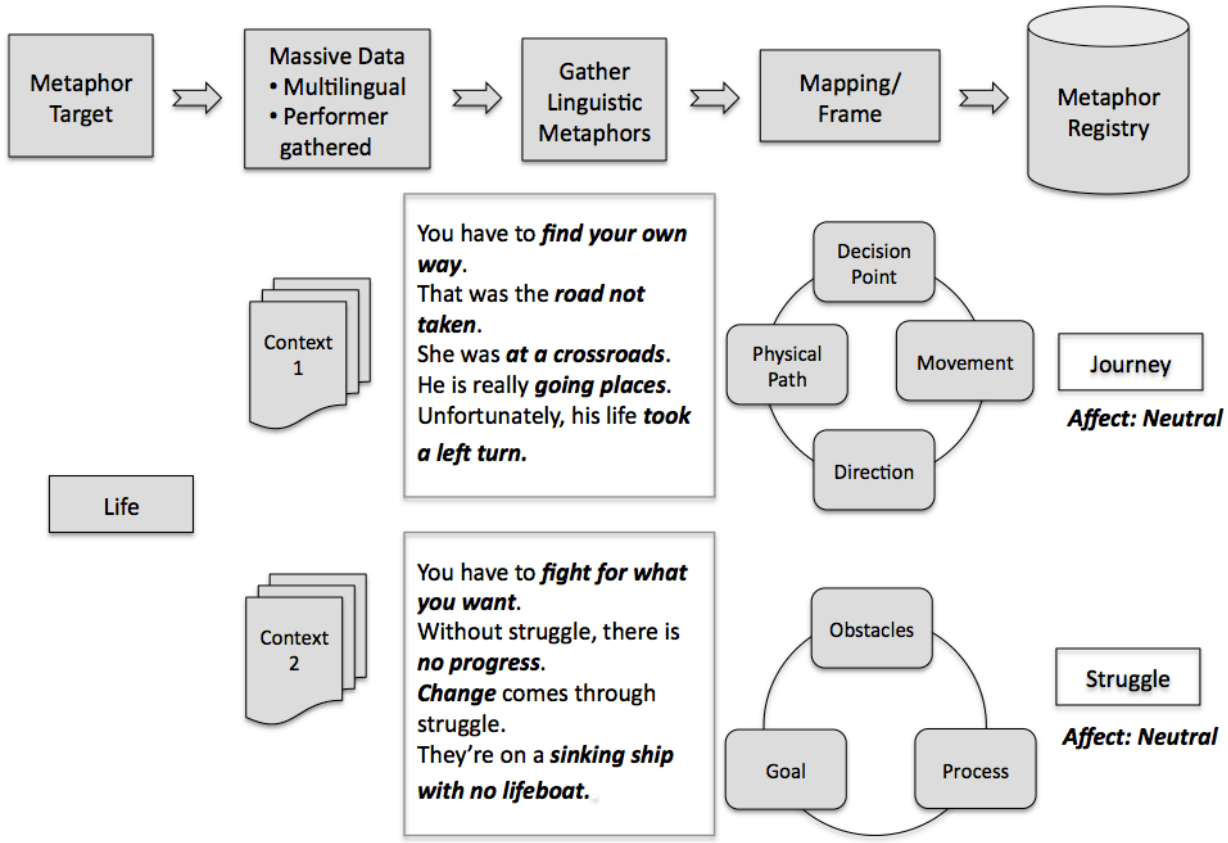

Figure 2-2. Notional data flow for automated identification, mapping, and classification of linguistic metaphors to conceptual metaphors (credit: IARPA)

IARPA has defined the steps required to automatically process metaphors within a large potentially culturally biased corpus (Figure 2-1) and has expressed interest in

associated advanced technologies and computational solutions. These steps are similar to the approach taken by Pasanek and Sculley to identify and classify metaphors in a large corpus [81]. An intelligence analyst would define the conceptual metaphor TARGET(s) of interest, the natural language processing would be responsible for identifying the corresponding linguistic metaphors contained in the specified corpus (Figure 2-2).

Metaphorical Criticism

Metaphorical Criticism, one aspect of Rhetoric Analysis, is the study of how specific writers and speakers use symbolic language as a persuasive mechanism to influence an audience. Metaphorical criticism is predicated on the notion that metaphors demonstrate social, cultural, and political views.

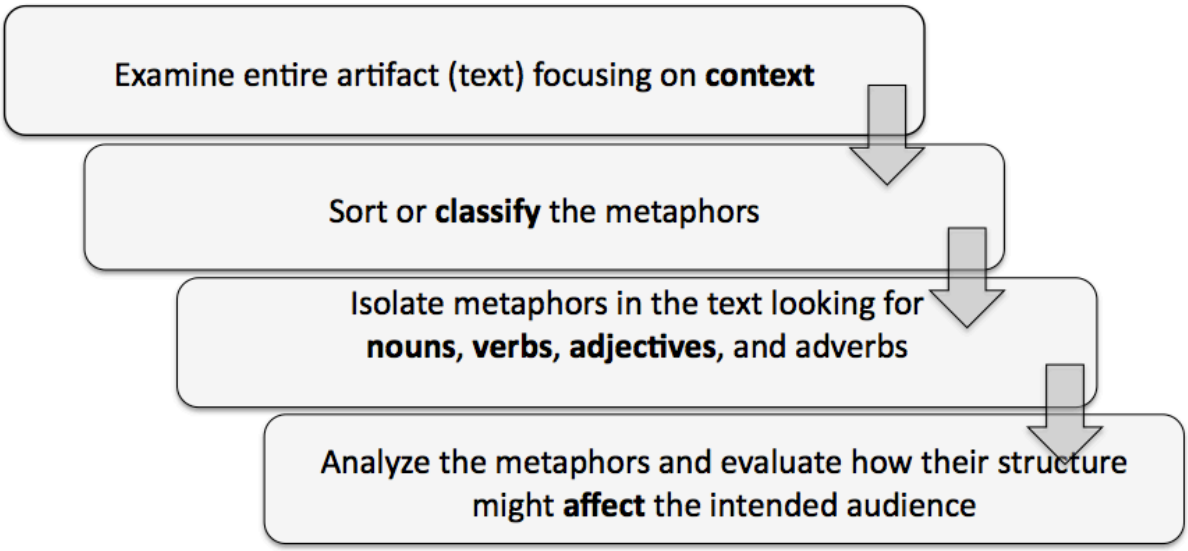

Figure 2-3. Metaphorical criticism processing flow steps

The process of metaphorical criticism breaks a text or given corpus into parts and then attempts to explain how the individual parts are composed to create a certain effect. The analysis identifies if the goal is to persuade, entertain, or educate the target

audience; identify the techniques employed and examples of the techniques; and the effectiveness and ineffectiveness of the SOURCE and TARGET domains utilized. A metaphor critic performs the steps by hand (Figure 2-3) on each text or source corpus attempting to identify the specific focus of the metaphor [40], what the metaphor implies, and if the metaphor is obscuring a concept or being omitted if a person is thinking in terms of the metaphor.

Mediation and Conflict Resolution

Psychologist, psychotherapists, mediators, and conflict resolution professionals all recognize that metaphors can be key to resolving or identifying the events and associated perceptions that are causing or contributing to a conflict. This approach is based on the premise that the way a problem is framed has a significant influence on the solutions that are available. Similar to a problem frame, conflict metaphors can be used to align individuals with a specific observation, question, or feeling [63].

Metaphors can assist antagonists and observers in understanding and communicating to others about things that have occurred or are occurring, thus framing events in a way that gives meaning to their perspective. In the scope of conflict resolution metaphors and the associated concepts can be divided into three categories based on their effect on conflict dynamics; positive, negative, and neutral. As such, a mediator can frame a conflict in a manner that may make it more likely to be transformed and resolved or more likely to polarize and escalate [63].

Mediators frequently use gardening, sports, games, and of course war metaphor categories. Unfortunately, metaphors do not convey the same concept or perspective to everyone even within a relatively small homogeneous population because metaphors are derived from social and cultural experiences. However, metaphors can also serve to

provide a new perspective to people from different cultures and social backgrounds. A mediator or psychotherapist would want to thoroughly examine all the written and verbal communications between the antagonists, focusing on the identification of metaphors within the context of the dispute, mapping the linguistic metaphors to conceptual metaphors, and then classifying them as having a positive, negative, or neutral effect on the issue and subsequently the resolution [63].

Why Metaphor Detection is Hard

The problem domain for identifying, mapping, and classifying metaphors is particularly hard since even humans hearing or reading the metaphor do not necessarily know the correct answer to what is being referenced. Metaphors are abstract and in many cases are very specific to the ideology, perceptions, and beliefs of the culture or language where they occur. Compared to other areas of computational linguistics where there is a quantifiably correct or incorrect mapping of the grammatical syntax associated with each sentence, metaphors present a more complex problem. Also, unlike humans, computers do not have the ability to apply high level semantic processing to languages and cannot apply background knowledge (experience) to the reasoning and inductive reasoning for the identification and interpretation of metaphorical language.

The automatic detection of metaphors can be divided into metaphor recognition and metaphor interpretation. Metaphor recognition is the ability to detect grammatical syntax that can be a metaphor and utilize the context to determine if a given sentence in the text is employing metaphorical language or is intended to be a literal statement. Metaphor interpretation is taking a metaphorical expression and identifying the intended literal meaning or a reasonable interpretation. Any algorithm intent on detecting or classifying a metaphor must account for a number of semantic features in addition to

the traditional grammatical features detected with the current natural language processing (NLP) pipeline. An NLP pipeline is a set of algorithms executed in a specific order where each algorithm adds syntactic or semantic information to the raw text.

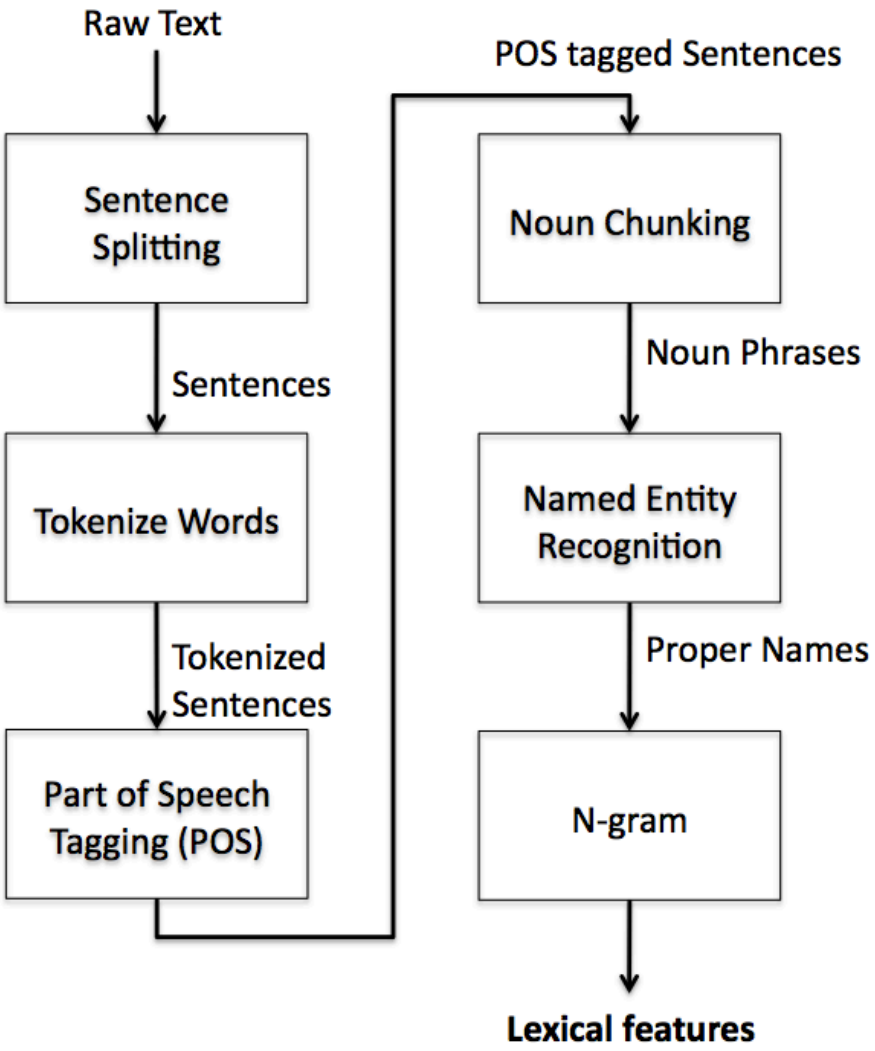

Figure 2-4. Natural Language Processing (NLP) syntactic feature pipeline for identifying and adding syntactic features to the individual sentences from a raw text source

The NLP pipeline shown (Figure 2-4) splits the raw text into sentences, each sentence into word and punctuation tokens, adds the part-of-speech (POS) tags based on probabilistic context free grammars (PCFG) [20][21], identifies the noun phrases,

performs named entity recognition, and creates an n-grams subsequence associated with each word token. In addition to the previously referenced syntactic processing another aspect of language that adds complexity is the use of a pronoun. This typically occurs where there is a grammatical relation between two words that have a common referent, specifically using a pronoun or similar word instead of repeating a word used earlier [87]. Table 2-1 contains a set of phrases and the resulting POS tags (Appendix A) and illustrates the ambiguity associated with pronouns and the difficulty a computer encounters while processing similar sentences.

Table 2-1. Example metaphor phrases and corresponding part of speech tags

Phrase	Part of Speech
He is a shark in a pool of minnows	He/PRP is/VBZ a/DT shark/NN in/IN a/DT pool/NN of/IN minnows/NNS
My lawyer is a shark	My/PRP$ lawyer/NN is/VBZ a/DT shark/NN
He is a snake	He/PRP is/VBZ a/DT snake/NN
She is a shrew	She/PRP is/VBZ a/DT shrew/NN
The sky is falling	The/DT sky/NN is/VBZ falling/VBG

The ambiguity is associated with the pronoun “he”, or “he who” if a pronoun refers to a human then the statement is most likely not literal. However, if ‘he’ refers to an actual male shark then the statement can be taken literally. Otherwise the phrase indicates a social relationship between an aggressive, assertive, or predator like individual (potentially a leader) and the other individuals as followers. The previous examples illustrate that the detection of metaphoric language requires more than just processing the grammatical structure to interpret if a sentence is literal or not, semantic information also needs to be included in any algorithm.

Semantics involves the study of the meaning of words, and phraseology and how these form the sentence meanings and the individual word lexical semantics. Individual word lexical semantics involves analyzing how individual words are related to one

another through the following semantic relationships hyponymy, hypernymy, antonymy and meronymy [87]. Where hyponymy and hypernymy describe the hierarchical semantic relationship where one word is a subordinate or part of a lower class of words forming an “is-a” relationship (e.g., superordinate = color, hyponyms = red, blue, pink) and meronymy forms “a part” relationship with another class of words (e.g., engine and axle are part of a car not a type of car). Antonymy defines words that are completely opposite in meaning, or mutually exclusive.

In addition to resolving the pronominal coreference, and the semantic relationships defined by hyponymy, hypernymy, antonymy, and meronymy the algorithms also need to take account of instances of polysemy. Polysemy is where words that are spelled the same have multiple distinct meanings (polysemes) and requires the algorithm to resolve which sense of a word is being used in the context of the sentence known as word sense disambiguation. Word sense disambiguation (WSD) is one of the more difficult problems in computational linguistics and involves algorithms that analyze the lexical semantics or the meaning of the words using hand coded knowledge in lexical dictionaries and thesauri [75], supervised machine learning methods for identifying classifiers [31][81][107], and unsupervised methods [88].

Challenges of Metaphor Detection

Using a machine to perform the processing of natural language presents a number of significant challenges. While a machine is significantly faster than a human parsing textual data and locating key words or named entities and providing the grammatical syntax metadata associated with the parse tree and POS tags, the identification and resolution of the lexical semantic features of a sentence and the variations in natural language make it difficult to detect. The detection of metaphorical language is heavily

dependent on the ability to determine the word senses within the context of the text, which requires metadata for both syntactic grammar features and the lexical semantic features.

Lexical Semantics

Any algorithm required to detect metaphorical language first needs the ability to identify the meaning of individual words and how the individual words relate to each other through lexical constructs such as hyponymy, hypernymy, and meronymy. The algorithm also needs to identify how words are combined together to form the meaning of sentences. The ability of an algorithm requires the use of a hand crafted lexical resources like WordNet [75]. While WordNet is a mature and quite comprehensive lexical dictionary it is still a hand crafted resource and suffers from the similar limitations associated with errors and omissions. The simple sentence “The sky is falling “ produces a syntax tree with the head noun (left most) containing “sky” and the rightmost verb phrase with the word “falling” shown Figure 2-5.

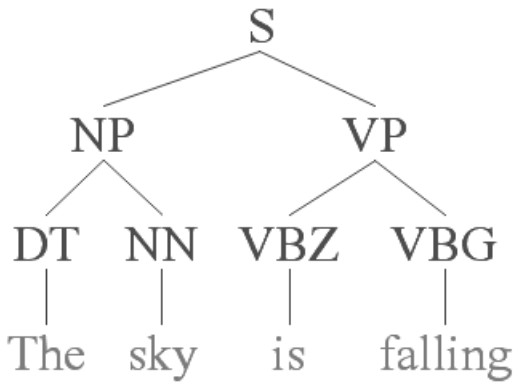

Figure 2-5. Syntax parse tree for sentence the sky is falling

Adding semantics information requires identifying the meaning of the noun “sky” and the verb “falling”. Using the lexical resources the hypernym chain can be determined from the lexical resources. Table 2-2 shows the hypernym chains or the

subordinate “Part-Of” relationships for the noun “sky” and the verb “falling” and that the chains do not intersect.

Table 2-2. Example hypernym chains for noun “sky” and verb “falling”

Sky (Noun)	Fall (Root Verb)
'sky.n.01'	'fall.v.01'
'atmosphere.n.05'	'descend.v.01'
'gas.n.02'	'travel.v.01'
'fluid.n.02'	
'matter.n.03'	
'physical_entity.n.01'	
'entity.n.01'	

Word Sense

The metaphorical language detection algorithm is also required to handle word sense disambiguation. Similar to the lexical semantics challenge, word senses need to be resolved either through one or more lexical resources that maps synonymy, homonymy, and polysemy for each individual word. Word sense resolution also requires the identification of the context of the text. Using the sentence “The sky is falling” and looking at the available word senses for the noun “sky” there is a single sense indicating the atmosphere of earth.

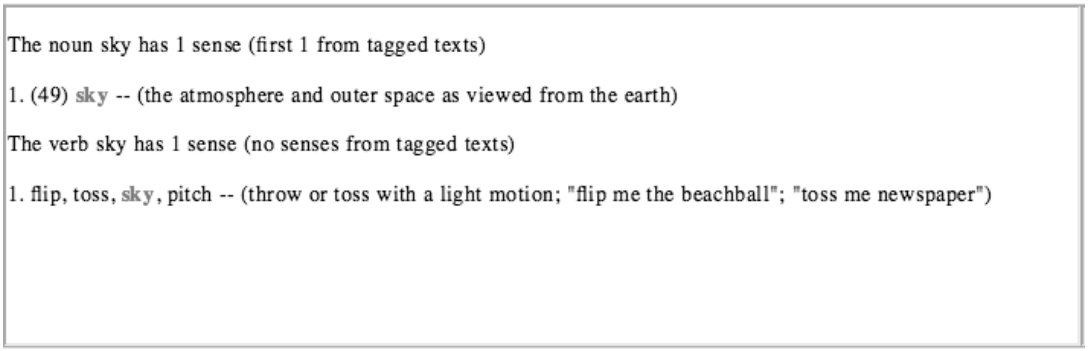

Figure 2-6. All senses for “sky” (output displayed in WordNet browser)

In addition the senses for the word falling also need to be evaluated. The syntactic processing and part of speech tagging can identify the grammatical features of the

individual words. However, there are 32 verb senses for “falling” and the root “fall” (Figure 2-7). Each word sense has a corresponding hypernym chain. Each word sense and associated hypernym chain is evaluated to determine the possible meaning of the word combination and identify if the sentence has a literal meaning or is a candidate metaphor.

The verb fall has 32 senses (first 15 from tagged texts)

1. (43) fall -- (descend in free fall under the influence of gravity; "The branch fell from the tree"; "The unfortunate hiker fell into a crevasse")
2. (21) descend, fall, go down, come down -- (move downward and lower, but not necessarily all the way; "The temperature is going down"; "The barometer is falling"; "The curtain fell on the diva"; "Her hand went up and then fell again")
3. (16) fall -- (pass suddenly and passively into a state of body or mind; "fall into a trap"; "She fell ill"; "They fell out of favor"; "Fall in love"; "fall asleep"; "fall prey to an imposter"; "fall into a strange way of thinking"; "she fell to pieces after she lost her work")
4. (6) fall, come -- (come under, be classified or included; "fall into a category"; "This comes under a new heading")
5. (4) precipitate, come down, fall -- (fall from clouds; "rain, snow and sleet were falling"; "Vesuvius precipitated its fiery, destructive rage on Herculaneum")
6. (4) fall -- (suffer defeat, failure, or ruin; "We must stand or fall"; "fall by the wayside")
7. (3) fall -- (die, as in battle or in a hunt; "Many soldiers fell at Verdun"; "Several deer have fallen to the same gun"; "The shooting victim fell dead")
8. (3) fall, shine, strike -- (touch or seem as if touching visually or audibly; "Light fell on her face"; "The sun shone on the fields"; "The light struck the golden necklace"; "A strange sound struck my ears")
9. (3) fall -- (be captured; "The cities fell to the enemy")
10. (3) fall -- (occur at a specified time or place; "Christmas falls on a Monday this year"; "The accent falls on the first syllable")
11. (3) decrease, diminish, lessen, fall -- (decrease in size, extent, or range; "The amount of homework decreased towards the end of the semester"; "The cabin pressure fell dramatically"; "her weight fell to under a hundred pounds"; "his voice fell to a whisper")
12. (2) fall -- (yield to temptation or sin; "Adam and Eve fell")
13. (2) fall -- (lose office or power; "The government fell overnight"; "The Qing Dynasty fell with Sun Yat-sen")
14. (2) fall -- (to be given by assignment or distribution; "The most difficult task fell on the youngest member of the team"; "The onus fell on us"; "The pressure to succeed fell on the youngest student")
15. (1) fall -- (move in a specified direction; "The line of men fall forward")
16. fall -- (be due; "payments fall on the 1st of the month")
17. fall -- (lose one's chastity; "a fallen woman")
18. fall -- (to be given by right or inheritance; "The estate fell to the oldest daughter")
19. accrue, fall -- (come into the possession of; "The house accrued to the oldest son")
20. fall, light -- (fall to somebody by assignment or lot; "The task fell to me"; "It fell to me to notify the parents of the victims")
21. fall, return, pass, devolve -- (be inherited by; "The estate fell to my sister"; "The land returned to the family"; "The estate devolved to an heir that everybody had assumed to be dead")
22. fall -- (slope downward; "The hills around here fall towards the ocean")
23. fall, fall down -- (lose an upright position suddenly; "The vase fell over and the water spilled onto the table"; "Her hair fell across her forehead")
24. fall -- (drop oneself to a lower or less erect position; "She fell back in her chair"; "He fell to his knees")
25. hang, fall, flow -- (fall or flow in a certain way; "This dress hangs well"; "Her long black hair flowed down her back")
26. fall -- (assume a disappointed or sad expression; "Her face fell when she heard that she would be laid off"; "his crest fell")
27. fall -- (be cast down; "his eyes fell")
28. fall -- (come out; issue; "silly phrases fell from her mouth")
29. fall -- (be born, used chiefly of lambs; "The lambs fell in the afternoon")
30. fall -- (begin vigorously; "The prisoners fell to work right away")
31. fall -- (go as if by falling; "Grief fell from our hearts")
32. fall, descend, settle -- (come as if by falling; "Night fell"; "Silence fell")

Figure 2-7. Word senses for the verb “falling” and root “fall” (output displayed in WordNet browser)

Context

Identifying the context of a particular sentence, paragraph, and textual document presents another challenge. The question is can the context of the document be learned

or inferred using the content of an individual text or corpus. Figure 2.8 shows the extensive graph for the root word “fall”

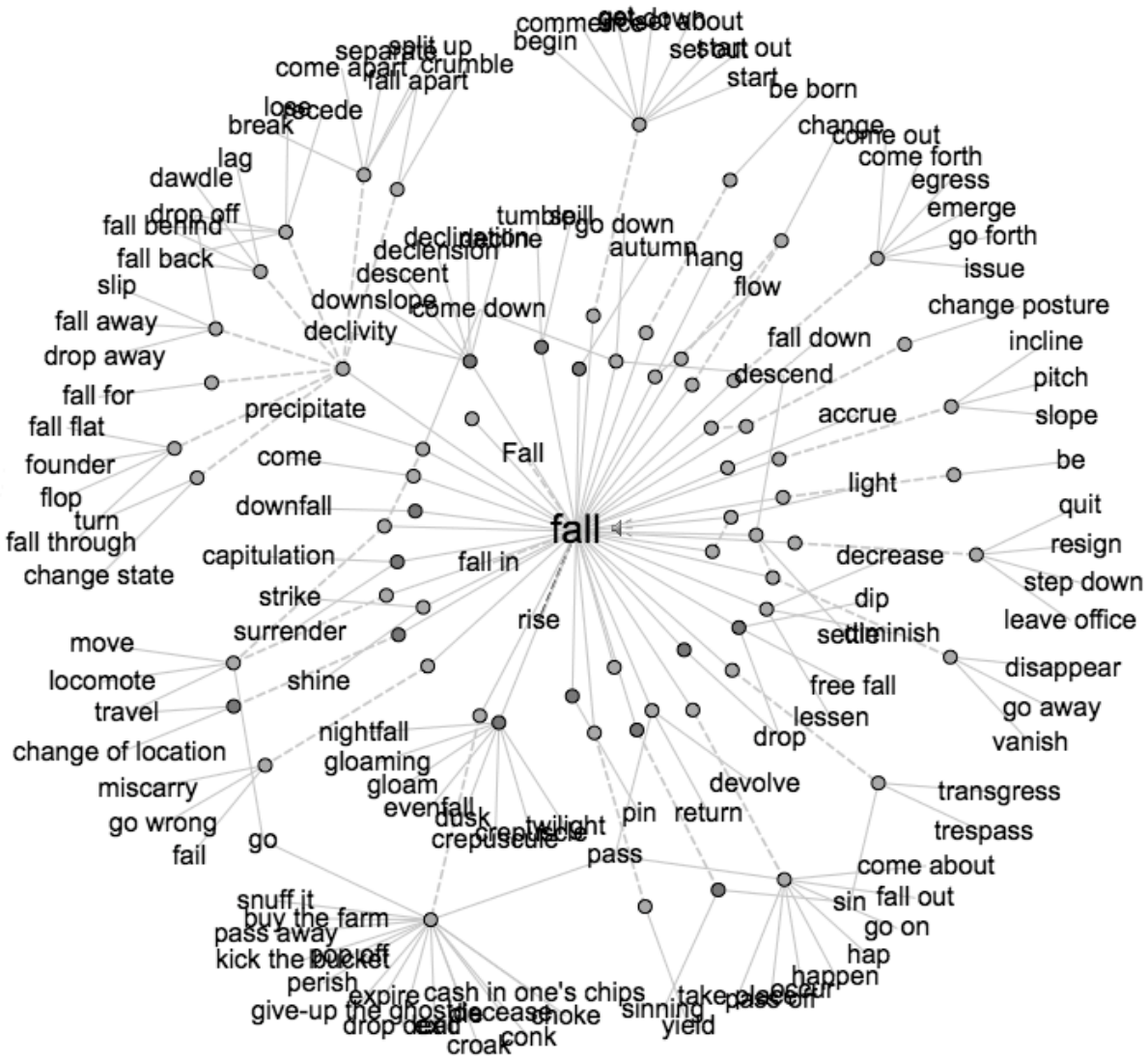

Figure 2-8. Full thesaurus graph of the word “fall” (output generated from VisualThesaurus[9])

Context at the document level can be determined by the source of the given text and metadata defining the genre (i.e., fiction, sports, etc). There are statistical methods for

---

[9] http://www.visualthesaurus.com/

determining topical context based on the words that are likely to co-occur given a target word sense. Local context requires information on the order of words and n-grams, along with order and inflection features.

Unsupervised Learning

Obviously a hands-on analysis of an individual text or corpus can be performed to identify the lexical features, word senses, and context features as well as identify and map metaphors. The hands-on technique is painstaking and time consuming [60][81]. One approach is the development of a training data set containing labels defined by hand and then perform a classification using a mainstream supervised algorithm such as Support Vector Machines (SVM) on the remaining texts in the corpus [81].

Ideally an unsupervised algorithm would be able to identify enough salient features about the lexical semantics, word senses, and context to perform a clustering of the features based on the assumption that similar senses occur in similar contexts and avoid the labor intensive efforts associated with manually identifying some or all of the lexical semantic metadata required to perform the detection of metaphorical language.

CHAPTER 3
BACKGROUND AND RELATED WORK

Metaphor Interpretation, Denotation and Acquisition System (MIDAS)

Martin [70] developed MIDAS as a computational approach to metaphor interpretation. MIDAS utilizes the knowledge representation language, KODIAK, to define a hierarchical organization of general conventional metaphors or core metaphors. KODIAK is an extended semantic network language similar to KL-ONE that connects knowledge elements through inheritance to form the hierarchical structure. MIDAS has been integrated into the Unix Consultant (UC), a program that is used to answer questions about Unix. Conventional metaphors are represented as associations between the SOURCE and TARGET concepts. MIDAS is composed of two subsystems; the Metaphor Interpretation System (MIS) and Metaphor Extended System (MES).

The MIS is responsible for processing each sentence to generate the syntactic and initial general semantic metadata. MIS augments the metadata with a set of concepts that provide a coherent explanation of the input sentence. These concepts may be the literal meaning of the sentence or an interpretation based on a conventional metaphor. MIS uses two approaches to determine the literal nature of a sentence, first checking if the abstract concept can be replaced by a concrete concept, and also by substituting a SOURCE concept with a TARGET concept.

The MES is executed when MIDAS cannot find conventional concepts for the input sentence. The MES includes three inference mechanisms; similarity–extension, core-extension, and a combination of the similarity and core inference mechanism. Similarity-extension follows the analogy principle associated with the traditional metaphor where there is a set of salience features that map the SOURCE and TARGET domains. Core-

extension assumes that a set of core associations in the SOURCE domain can be mapped to the TARGET domain. The MES algorithm attempts to find the match of an existing metaphor in the knowledge base to interpret the input sentence. New metaphors interpretations are added back into the knowledge base.

MIDAS is predicated on the assumption that new metaphors can be derived from existing metaphors. The core conventional metaphors are hand coded, and MIDAS is unable to detect/learn metaphors that are not related to the core metaphors by inheritance. Martin identifies MIDAS as being limited by size and correctness of the knowledge base of non-literal conventions and the knowledge base has no real coverage and is not empirically verifiable [71].

Met* (met star)

Fass developed met* [37] which is a method for computationally discriminating metonymy and metaphor implemented in the met5 system. The met* method is part of Collative Semantics (CS) [36][35] a domain independent semantics for NLP similar to frame [38][39] or semantics based networks. CS is intended to address lexical ambiguity including metaphorical relations. CS utilizes knowledge representations and knowledge structures for matching/mapping results into and semantic vectors. In CS there are seven types of semantic relations defined: literal, metonymic, metaphorical, anomalous, redundant, inconsistent, and novel relations. CS extends the concepts defined by Fass and Wilks known as Preference Semantics (PS) [34]. PS is the underlying strategy employed for passing English into a semantic representation without an explicit model devoted to syntactic analysis and without classification of the words into their parts of speech (i.e., noun, verb, adjective, and adverb). The second step in

the CS involves identifying word senses and the creation of sense frames containing pairs of word senses. The met* method is described in the flow chart in Figure 3-1.

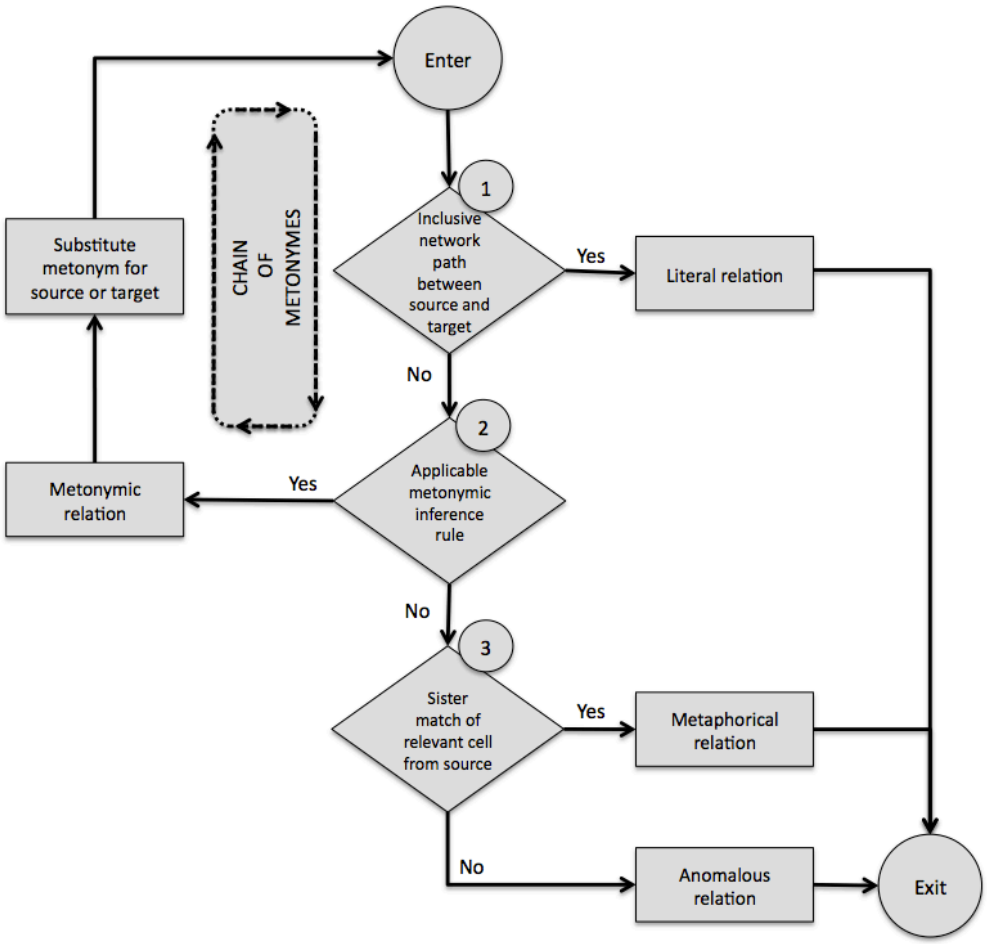

Figure 3-1. The met* method with Collative Semantics (credit: Fass 1991)

In sentences having the traditional metaphor form X is (like) Y the met* process first checks if X and Y satisfy selectional restrictions. Fass identifies a significant problem with selectional restrictions in that there are well-formed sentences that have a metaphorical interpretation and contain no violations. If no restriction is found then X is (like) Y contains a conflict. Met* first checks for metonymy and if there is not suitable metonym model then met* invokes the metaphor search. If the metaphor does not find a metaphor then the statement X is (like) Y defined as an incorrect statement.

One key to met* is that the verbs are represented by a semantic vector where the elements of the vector are selectional restrictions for the verb sense. The representation emphasizes a linear property of language rather than a recursive one by breaking language into phrase and clauses rather than sentences. This uses inference and a progressively deepening best-fit algorithm approach. The following examples show that transitive and intransitive verbs play a significant role in metaphors. They indicate a specific dictionary where contextual information is defined in the semantic formulas for verbs. Polysemy rules are applied to metaphors involving predicates and non-predicates.

- The car drank gasoline: only a verb sense the most common is take in liquid
- The car drinks gasoline: has noun senses and verb senses

The word car is a noun and there is no verb sense. The last example we see how even a grammatically correct sentence can be difficult to determine the word senses further illustrating that metaphors are not really a binary (true/false) matter, but rather a representational satisfaction. This illustrates that word sense is important in both the syntactic and semantic notion.

Met* leverages context, world knowledge, and analogy inference to identify metonymy and metaphor in a given sentence. However, Fass emphasized that the met* method had only been applied to a small set of English sentences. This is primarily due to limitations associated with the scope of the hand coded ontology and selectional restrictions.

Selectional Preferences

Most verb instances prefer a particular type of argument. These regularities are called selectional preferences or selectional restrictions [67]. One example is the verb

‘eat’ tends to have an object that is a food item, another example where the subject for ‘bark’ tends to be dogs. These are referenced as preferences as opposed to rules because they can be overridden in metaphorical language. The ability to acquire selectional preferences in statistical NLP is important because machine-readable dictionaries such as WordNet [75] could be missing a number of words found in a given corpus, allows the meaning to be inferred. Resnik [88] proposed a selectional preference model that can be applied to any class of word that imposes semantic constraints on a grammatically dependent phrase: noun-noun, adjective-noun, verb-noun, verb-direct object, verb-subject, and verb-prepositional phrase.

The model uses two notions to formalize selectional preference strength and selectional associations. Selectional preference strength measures how strongly the class of words constrains another class of words. Selectional preference defines strength $S_R(p)$ as follows:

$$S_R(p) = D_{KL}(P(C \mid v) \parallel P(C)) = \sum_c \Pr(c \mid p) \log \frac{\Pr(c \mid p)}{\Pr(c)}$$

Where Pr(*C*) is the overall probability distribution of a word classes and Pr(C|v) is the probability distribution of a word classes in the direct dependency position *v*. For probability distributions *P(C|v)* and *P(C)* of a discrete random variable the Kullback–Leibler (K-L) divergence [59] or relative entropy is defined to be:

$$D_{KL}(P(C \mid v) \parallel P(C)) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

Where it is the average is taken using the probability *P(C)* of the logarithmic difference between the probabilities of *P(C|v)* and *P(C)*. If *P(C|v)* and *P(C)* both sum to 1 and if *Pr(c)*>0 for any i such that *Pr(c|p)*>0 then the K-L divergence is defined. Based on the

selectional preference strength the selectional association between verb *v* and a class *c*.

$$\lambda_R(p,c) = \frac{1}{S_R(p)} \Pr(c \mid p) \log \frac{\Pr(c \mid p)}{\Pr(c)}$$

Typically, selectional preferences and selectional associations are calculated for verbs, that is, the verb is said to select for its direct object, subject, indirect object, etc. For example, the verb “eat” selects for “food” as its direct object. This approach was defined by Resnik in [88] and effectively applied by Mason in the CorMet [73] for identifying verb preferences for each noun contained in a corpus. Baumer also applied selectional preferences in his Computational Metaphor Identification (CMI) [8] where he identified the preferred noun for each verb in the corpus. One primary difference between the CorMet [73] and CMI approaches [8] is the focus on verbs versus nouns respectively. Characteristic verbs are indicative of the types of actions and relations described in a corpus, but the characteristic verbs do not necessarily select for characteristic nouns.

Corpus-Based CorMet

Mason developed CorMet [73] that is a corpus-based system for creating a metaphor mapping between concepts. CorMet learns the selectional preferences of the domain specific characteristic verbs by analyzing large corpora of domain specific data. Mason points out that most computational models for metaphors are dependent on hand-coded knowledge bases and only work on a very narrow set of domain specific examples. Although CorMet is designed to leverage the WordNet lexical dictionary and work on a larger scope than other systems such as MIDAS [70] and met* [37], Mason states “Note that CorMet is designed to detect higher- order conceptual metaphors by

finding some of the sentences embodying some of the inter-concept mappings constituting the metaphor of interest but is not designed to be a tool for reliably detecting all instances of a particular metaphor" [74].

CorMet uses the k nearest neighbor (kNN) clustering algorithm to identify clusters around WordNet nodes facilitating a more accurate identification of selectional preference similarity. As part of CorMet mapping the polarity is calculated between the two concepts or two domains by measuring the magnitude and direction of the structure transfer. A confidence measure is computed for each metaphor that CorMet discovers. The confidence is based on the evidence about a specific metaphor into one number, but this is not intended to be the probability as there is insufficient data and the confidence measure is actually arbitrary.

Mason‘s approach includes a machine learning technique based on the number of verbs mediating a metaphor, CorMet decomposes all the words in a corpus to a bag of words representation, stems the words, and finds the ratio of occurrences of each words stem in the total number of stems. CorMet selectional preference learning is intended to be error tolerant since complex sentences are typically parsed incorrectly. Mason applies the selectional preference algorithm defined by Resnik [88] to perform sense disambiguation and then clustering over the results using kNN (where k = 1).

CorMet is tested on a subset of the metaphors defined in Lakoff's Master Metaphor List [62], focusing specifically on the higher level metaphors. Due to the version of WordNet Mason used in the experiment, CorMet is biased toward verbs with common nominal homonyms within the corpus domain.

Computational Metaphor Identification (CMI)

Baumer developed CMI [8], a system that is capable of identifying metaphors in written text. CMI maps selectional preferences of frequent nouns found in a source corpus to a target corpus where the mappings indicate potential metaphors. CMI processing consists of four primary steps: corpus preparation, source suggestion, finding mappings, and metaphor presentation. Corpus preparation is performed on any potential source or target corpus. Source suggestion is an optional intermediate step where CMI compares a prepared target corpus with several potential prepared source corpora and suggests to which source corpora metaphors should be mapped. The source suggestion provides an indication of the types of metaphors that may be employed in a target corpus identifying SOURCE domains. Finding mappings identifies potential metaphors from a prepared source corpus to a prepared target corpus.

Corpus preparation Involves the acquisition of a desired corpus followed by the NLP processing required to determine the parts of speech and subsequently identify the characteristic nouns (most frequently used), learn the selectional preference for each noun, and lastly the clustering of the WordNet synsets associated with each noun. Parsing enables the identification of the grammatical relations (i.e., subject, object, adjunct, compliment) for each word in a sentence along with the average number of grammatical relations per sentence. CMI takes the approach of finding the characteristic nouns. This approach is predicated on the assertion that the SOURCE and TARGET domains of a metaphor map contain salient characteristics. The salience characteristics as identified by Gentner [43], Ortony [79], and Weiner [103]. This deviates from the CorMet approach [73] of finding relatively frequent verbs. Baumer identified an issue with the CorMet use of verbs and WordNet based approach is subject to nominal

homonyms where a word can have both noun and verb senses. Characteristic nouns are nouns that occur with a relatively high frequency in a given corpus. Note that WordNet also provides relative frequency of use as an attribute for each sense of a word, however, Baumer does not use this a comparator for characteristic nouns.

Selectional Preference Learning is responsible for calculating the selectional preferences and selectional associations [88] for each characteristic noun identified in the corpus. CMI trims the list of characteristic nouns using custom list of stop words that includes pronouns, prepositions, some common verbs, and other words with nominal homonym. Baumer asserts that this approach is significantly less rigid than the method using heuristic templates employed by Mason in CorMet. Both CorMet and CMI leverage selectional preference sparse data vector clustering with the k-Nearest Neighbor (kNN) algorithm, however, CorMet uses two iterations of nearest one neighbor and CMI uses one iteration of two nearest neighbors clustering. Baumer points out that in CorMet Mason utilized the dot product as the similarity measure associated with each iteration of the algorithm. However, while analyzing the Wikipedia data, Baumer discovered that the dot product was an unsatisfactory similarity measure. This was based on the scenario where the calculation rewarded similarity without penalizing dissimilarity resulting in clusters of mostly unrelated synsets centered around the ancestors of many of the synsets.

Baumer notes that in some cases selectional associations of conceptually similar synsets are surprisingly different. To address this, synsets are clustered based on their selectional associations for the verbs for which they select. Each synset is represented

as an n-dimensional vector, where the nth element is the synset's selectional association for the nth verb-case slot.

Once a source and target corpus are selected and prepared CMI is responsible for finding potential metaphor mappings by calculating selectional associations for each cluster. In CMI a cluster's selectional associations are calculated as the average of the selectional associations for all the synsets in the cluster. CMI employs two weighting factors: 1) the polarity of mappings between two clusters as a weighted sum of the selectional associations for all verb-case slots selected by both clusters, 2) between the CMI weights the target domain higher with respect to relative frequency of the verb this is based partially on the notion of salience imbalance between the SOURCE and TARGET domains [79][103]. CMI uses different weights for the SOURCE and TARGET domains, .25 for the SOURCE domain and .75 for the TARGET domain based on the concept of salience. CMI then applies a logarithmic scaling with the assertion that this approach helps ensure identified metaphors are derived from the most meaningful verbs.

CMI was embedded in a larger context to assist in the learning and retention of students by helping them identify a metaphor for a given concept. The success of CMI is predicated on the ability to identify metaphor mapping and presenting them to the student as a clue/key for learning a particular new concept. As Baumer points out there are a number of tuning parameters that were based on the results given the particular corpus. He also points out that there was no real benchmarking performed with respect to accuracy on metaphor detection, therefore, while it appears that CMI is an

improvement on CorMet particularly in the area of needing to define a metaphor of interest there is minimal evidence to support this assertion.

Trope Finder (TroFi)

Birke and Sarkar [11] define a system for automatically classifying literal and non-literal usages of verbs through nearly unsupervised word-sense disambiguation. TroFi utilizes sentential context instead of selectional constraints violations or paths in semantic hierarchies. Similar systems can be separated into rule based and dictionary-based systems. Rules based systems are employed by a number of the metaphor and metonym processing systems [37][70] where the rules must be largely hand coded or confined to a very small domain. Dictionary based systems utilize path distances between the words to detect figurative language. Corpus based systems basically extract or learn the metaphor processing information from a large corpus avoiding the need for manual annotation or metaphor map, and rule creation. Similar to CorMet [73], TroFi uses contextual information acquired from a large corpus and also uses WordNet as a primary knowledge source and lexical dictionary, but TroFi does not use selectional preferences.

Mapping Principle (MP)

Ahrens [3] defines Mapping Principles for conceptual metaphors using WordNet and Suggested Upper Merged Ontology (SUMO) [19]. The MP are followed with a corpus based approach to determining the salience between SOURCE and TARGET domain pairings [2]. The integration of the Conceptual Mapping Model (CMM) with an ontology based knowledge representation (SUMO) to demonstrate conceptual metaphor analysis can be restricted (constrained) and eventually automated.

Two approaches are utilized to verify the mapping principles; 1) the first approach is a frequency of occurrence in the corpus for the conceptual metaphors associated with ECONOMY, 2) the second approach uses the lexical linking between WordNet and SUMO[10] to delimit the source domain knowledge [4][18][19]. Leveraging the inference and constraints built into the ontology by [78] Ahrens proposes to look at the WordNet sense, the WordNet definition, and the SUMO node for the WordNet sense associated with the intuition-based examples. This approach is needed in order to determine if any overlapping semantic coverage contained in these three types of information. Ahrens identifies that there was a significant manual effort required to define the mappings. Ahrens proposes a method to reduce the subjectivity and possibly use the WordNet senses and compare the level of abstraction.

Attribute Metaphor (ATT-Meta)

Barnden and Lee, et el. have implemented the Attribute Metaphor (ATT-Meta) [6] system to perform some of the reasoning and inference required to computationally understand metaphorical expression. They report that ATT-Meta can perform the basic processing of metaphors and correlate some of the metaphor instances with the Master Metaphor List defined at Berkeley [62]. ATT-Meta demonstrated at a primitive level that the system was capable of handling metaphorical utterances based on the main metaphorical views provided by the Master Metaphor List. The metaphor examples used in the experiment were manually extracted from a large corpus [7].

[10] Suggested Upper Merged Ontology (SUMO) http://ontology.teknowledge.com , Niles and Pease, 2003

Mining Millions of Metaphors

Pasaneck and D. Sculley utilized a database, ‘The Mind is a Metaphor” [82], of metaphors that were mined by hand over number of years using keyword searches and other mechanisms for electronically collecting text. These metaphors were categorized as ‘heart’, ‘mind’, and ‘soul’ and were utilized as the training set. The authors then used a supervised learning approach that consisted of a linear Support Vector Machine (SVM). The classifiers were trained to recognize metaphors from non-metaphors using portion of the hand coded metaphor database. Then mapping the known metaphors to vectors, using the bag-of-word model. This model represents the corpus as an unordered collection of words and shows the number of occurrences of a word in the given corpus.

The authors sanitized the corpus by making all words lower case and all punctuation had been removed. The authors reported an F-measure accuracy of .867 across multiple authors[11]. The accuracy number is state of the art for natural language processing and illustrating that given a training set that is sufficiently large and broad enough to cover multiple authors, however, the use of the bag-of-words approach is not utilizing the lexical semantic relations that help define metaphors within a context. That means this approach is heavily dependent on a clean training set and is therefore less desirable from an automated semantic enrichment and extraction of complex phrase/figures of speech such as metaphor.

[11] Full statisitcs are available in Table 1 on page 11 of Mining Millions of Metaphors [81]

What are the Missing Pieces?

The main focus for most of the research falls into either detection of metaphors in a given text and interpreting metaphors. There is also a focus on classifying metaphors based on the hierarchical taxonomy of conceptual metaphors defined in Lakoff ‘s Master metaphor list [62]. The primary focus in this research is to determine how to increase the confidence that a metaphor was detected in a given text using mainstream machine-readable dictionaries available as the lexical resources used in unsupervised manner, not necessarily to increase the number of metaphors detected, but to reduce the number of false positives.

We have observed several knowledge driven approaches for both detection and interpretation of metaphors depending significantly on hand coded knowledge, associations, and linguistic rules. These approaches have typically been limited to very small domains in proprietary modeling languages (e.g., KL-ONE as used in Met* [37]) making them difficult to scale beyond the original sample domain. There has been some recent work that utilizes open world knowledge sources WordNet and SUMO. These resources are still built and maintained by hand and are in no way considered completely comprehensive, however, they have the distinct advantage of being developed as a more general purpose knowledge bases as opposed to the smaller examples we have observed.

Ideally a metaphor detection algorithm can leverage these lexical resources as needed in a manner similar to the way a human will apply personal experience and other background knowledge to detect and interpret metaphors in written communications. In the absence of experience in a particular linguistic area a person is likely to search for information using dictionaries, thesauri, encyclopedias, and online

through one of the numerous search engines depending on the resources that are available. However, each these resources do not have the same level of pedigree, specifically is information found in a dictionary more credible than the results found on an individuals online blog.

While there has been research that utilizes the information contained in the lexical resources to identify word senses and resolve a number of the lexical semantic relations utilizing unsupervised clustering algorithms, very little research has been published in the area assigning weights or confidence levels to the many lexical resources and the role they play in word sense disambiguation. The distinguishing factor here is that our project's goal was to improve the confidence that a metaphor candidate was detected based on the lexical resources that provide supporting semantic information, including identification of word senses and lexical semantic relationships, but also taking into account context at the given text and corpus level and any conflicting information from the lexical resources.

We have shown that there are sentences that can represent both a literal statement and a metaphor making metaphors something other than a yes or no. This means there is a level of ambiguity that needs to be accounted for when identifying metaphor candidates. The question remains of how to detect a metaphor candidate that satisfies the some or all of the rules that define a traditional or linguistic metaphor. Our research is focused on detecting metaphor candidates in a given text and assigning a confidence weight based on the salience of characteristics that relate the words that could compose the SOURCE and TARGET domains. Where the salience characteristics are defined as lexical semantic relationships (e.g., polysemy) that are

detected using an algorithm that accesses one or more lexical resources, and each of lexical resource, the pedigree of the lexical resource. The pedigree is a weighting factor defined by the maturity of the lexical resources, coverage of the lexical semantic attributes, and word senses.

Contribution

Innovation/Value Added

The detection of metaphorical language can be highly complex, computationally expensive, and explicitly tied to a context or domain. The grammatical syntax parsing of a sentence provides insufficient information to determine the literal nature of an utterance. The lexical semantic features of the sentence must also be identified including the determination word sense and along with context evaluation must be performed on each sentence. Large knowledge bases and domain rules currently are constructed by hand making them brittle and difficult to scale for large use. The corpus based statistical approaches, such as selectional preferences, are only a partial solution as they are applied to a specific dependency between noun – verb.

The application of unsupervised learning methods are being used for the detection of lexical semantics as well as word sense disambiguation have been performed with a measure of success. Evaluations performed as part of SensEval[12] are indicating precision and recall number in the range of 49–51%, while grammar and syntax based methods are delivering performance in the 89–95%[13]. The detection of metaphors relies heavily on the ability to determine the word sense, the lexical semantic relationships,

[12] Sens Eval http://www.senseval.org/ is an evaluation exercise for the Semantic Analysis of Text sponsored by ACL-SIGLEX

[13] SensEval-2 Results: http://86.188.143.199/senseval2/Results/all_graphs.htm

and the context of the given text. The unsupervised statistical approach does not take into account the context of a given text. Instead, the only relationship that is calculated is from what one word has with another using a lexical resource to determine the relationship. This makes them highly dependent on the machine-readable lexical resources to provide/identify the lexical semantic relationships between the words.

Somewhere between the approaches that use knowledge base lookup/search, the rote learning or memorization approaches, and the unsupervised statistical approaches there is an approach that can leverage the best features of all while minimizing the drawbacks. For this hybrid approach to be effective it must account for the context of the text, utilize the lexical knowledge bases and the statistically defined relations to resolve the word sense and lexical semantic features needed to evaluate if the sentence is literal.

This method employs a unique approach of pattern recognition, artificial intelligence techniques, using heuristics (similarity/weighting factors), statistical model for clustering, and natural language processing to provide a confidence factor for each candidate metaphor and candidate rules for the knowledge base. For the sentence *"*The senate has become the battleground for health care lobbyists.*"* is a 'WAR' metaphor, where the implication is that opposing lobbyist are attempting to seize territory from the Senate.

This sentence is first tokenized and parts of speech added to the metadata. From the part of speech tags we can determine the noun and verb phrases. We also apply the traditional metaphor heuristic X is (like) Y since the detection of this construct increases the probability that the sentence contains a metaphor. From the nouns and

verbs detected we can establish a relationship, and use the statistical method, selection preferences, and a clustering method (e.g., k Nearest Neighbor) to determine if the noun and verb classes have an affinity for one another within the general corpus. The more frequently used nouns and verbs can indicate the context since a given text is likely to contain more than a single word that is related to the context.

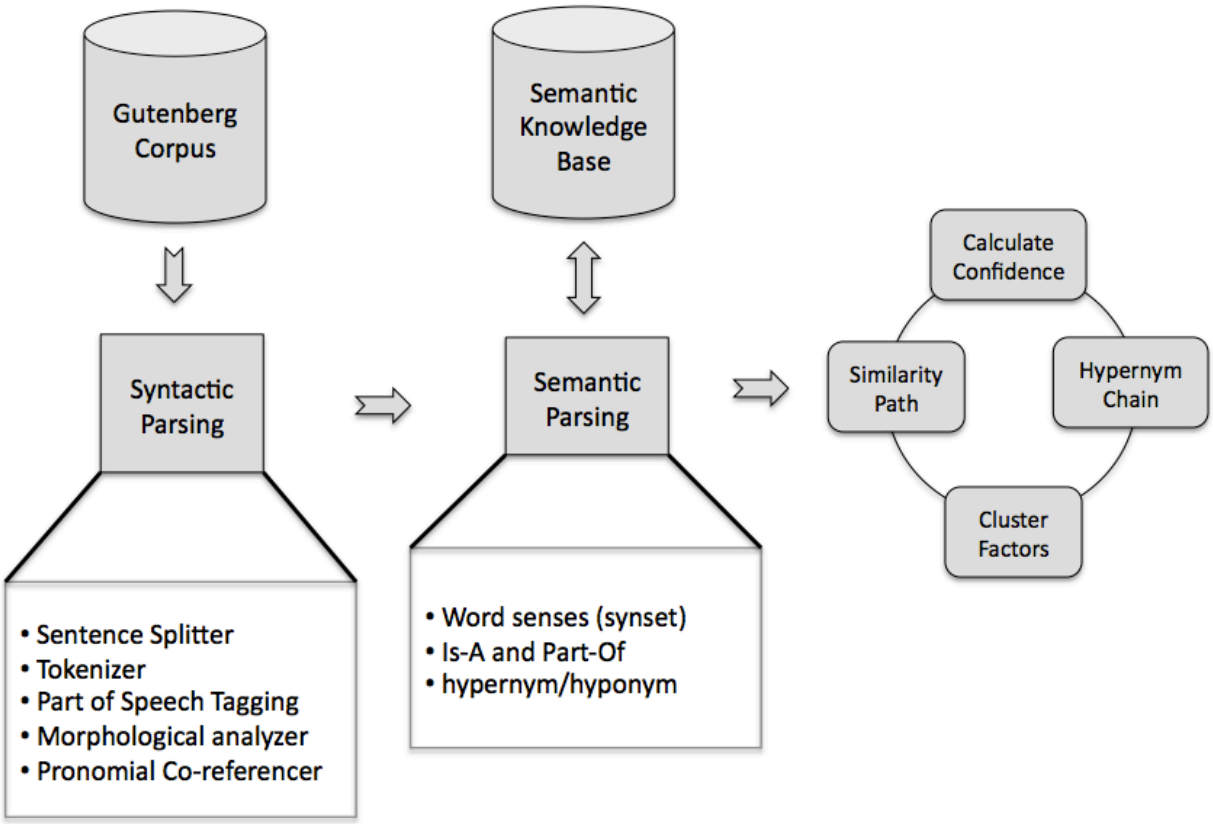

Figure 3-2. High-level algorithm processing flow, starts with a corpus, parses the text to identify grammatical metadata, processes the nouns and verbs to determine dependency relationships, and measure for lexical semantic weighting

If the nouns-verb combinations are infrequent then this increases the probability that this is a metaphor. Some other heuristics are the check for synonymy and hypernomy – using a lexical resource we can begin to calculate a confidence factor based on a probability calculated from the checks with lexical semantic attributes of X

and Y (i.e., is X a synonym of Y, is X a hypernym of Y, is X a polyseme for Y). Figure 3-2 illustrates the high level processing flow and indicates where semantic metadata is added.

Contributions

The prototype developed to perform the steps identified (see Figure 3-2) and collect the results will most likely become shelf ware. However, this research is not without value. This research crosses several computer science disciplines. The primary classification hierarchy defined by the Association for Computing Machines (ACM) Computing Classification System (CCS). The areas covered include I.2.7 Artificial Intelligence/Language parsing and understanding, I.2.8 Artificial Intelligence/Heuristics, and I.5.3 Pattern Recognition/Algorithms. These classifications are illustrated in Figure 3-3.

The problem space domain is I.2.7 Artificial Intelligence/Language parsing and understanding – in this case, parsing the text that contains parts of speech metadata, adding lexical semantic metadata for each word, and searching for metaphorical utterances. The first contribution from this research is the automated detection of metaphor candidates, as previously defined identification of a metaphor is not a true or false. There are a number of factors that must be examined and calculated to determine a probability that the sentence contains a metaphorical utterance. Figure 3-4 shows the specific subjects in the I.2.7 classification.

The solution space domain related to this research includes both I.2.8 (AI/Heuristics) and I.5.3 (Pattern Recognition/Algorithms) shown in Figure 3-5 and Figure 3-3 respectively. These are the areas that were leveraged to address the problem solution – how to detect metaphor candidates.

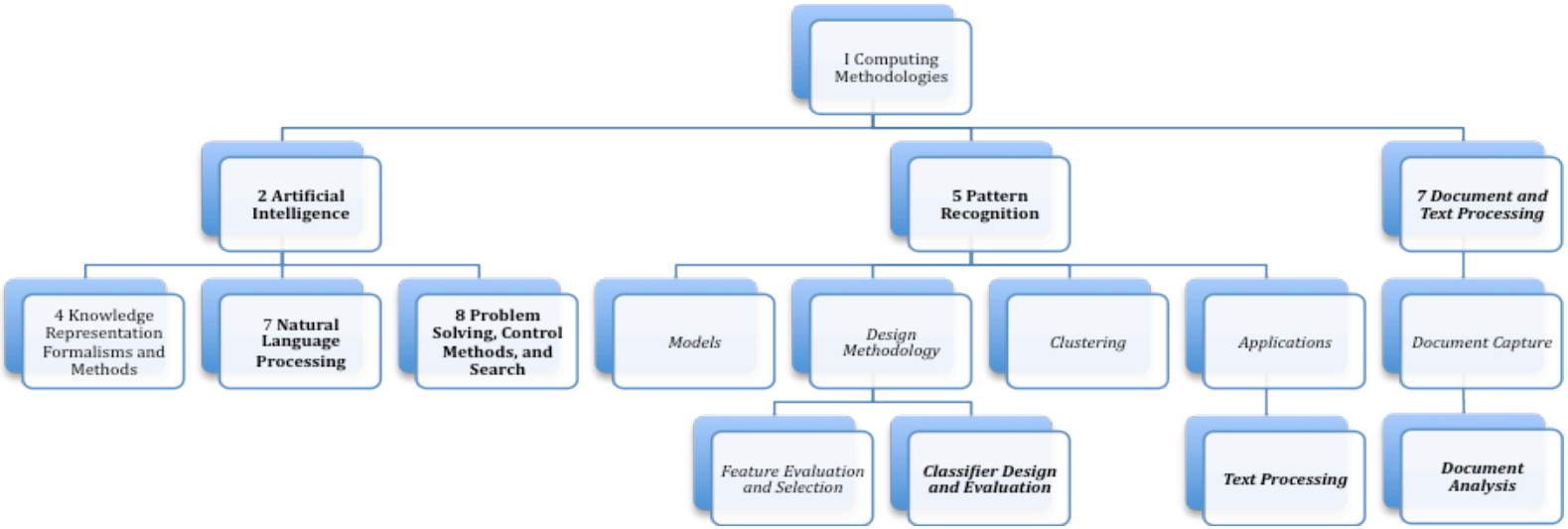

Figure 3-3. Relative contribution within computer Science

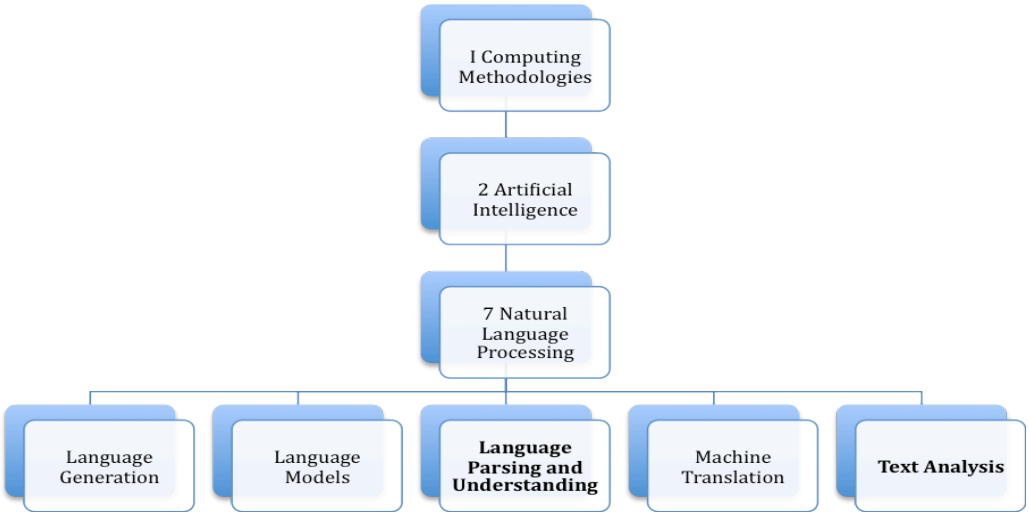

Figure 3-4. Relative contribution within Natural Language Processing

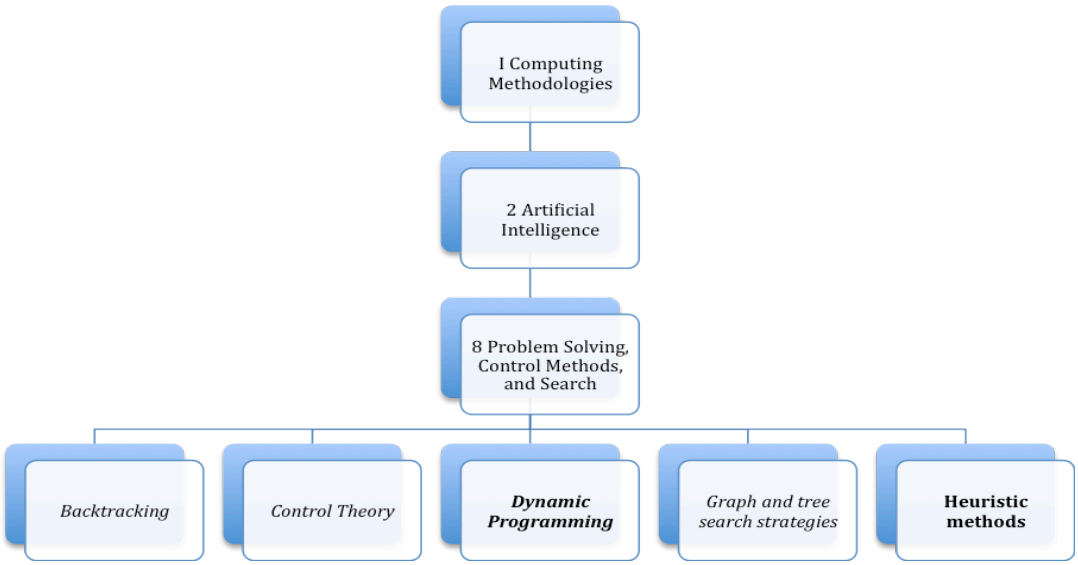

Figure 3-5. Relative Contribution within problem solving, control methods

CHAPTER 4
PROTOTYPE

The research into an effective mechanism for the detection of metaphorical utterances in a given sentence resulted in a prototype that ingests a corpus, processes each sentence in the corpus both grammatically and semantically, and calculates a confidence factor based on semantic relationships detected. Prototype software was developed to perform several experiments on a general corpus to detect traditional and linguistic metaphor candidates. The prototype is designed in a loosely coupled manner allowing for the machine-readable dictionary (i.e., WordNet) to be augmented with additional lexical/semantic resources such as VerbNet or FrameNet. The prototype development was accomplished in two phases; 1) detection algorithm using literary term heuristics and basic hypernym chaining using rule grammars to group grammatically similar sentences, 2) extended detection algorithm including word similarity based on semantic relationships and identifying salient characteristics that can be leveraged in clustering algorithms such as k-means and k-NN.

Corpus

The experimental results and corresponding analysis are based on experiments resulting from applying the software algorithms to detect metaphor candidates to a domain independent corpus downloaded from the Project Gutenberg [48]. Project Gutenberg is a freely available online corpus of 36,000 books in a machine-readable simple text format. The corpus of books is assembled from a broad spectrum of genre including children’s stories, fiction, poetry, classical literature, and technical reports.

Lexical Semantic Resource

The machine-readable dictionary resource identified for use in the prototype is WordNet version 3.0. WordNet [75] is a freely available, large lexical database of English nouns, verbs, adjectives and adverbs grouped into sets of cognitive synonyms called synsets. WordNet also labels the semantic relations among words. Tables 4-1, 4-2, and 4-3 contain the words, synset, senses, and polysemous statistics for WordNet[14].

Table 4-1. WordNet 3.0 Number of Words, synsets, and senses

POS	Unique Strings	Synsets	Total Word-Sense Pairs
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580
Total	155287	117659	206941

Table 4-2. WordNet 3.0 Polysemy information

POS	Monosemous Words and Senses	Polysemous Words	Polysemous Senses
Noun	101863	15935	44449
Verb	6277	5252	18770
Adjective	16503	4976	14399
Adverb	3748	733	1832
Total	128391	26896	79450

Table 4-3. WordNet 3.0 Average Polysemy information

POS	Including Monosemous Words	Excluding Monosemous Words
Noun	1.24	2.79
Verb	2.17	3.57
Adjective	1.40	2.71
Adverb	1.25	2.50

Each of WordNet's 117,000 synsets defines a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. In many ways WordNet

[14] http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html#toc3

resembles a thesaurus in that words are grouped together based on their meanings. WordNet interlinks word senses as well as word forms resulting in words that are found in close proximity to one another in the network are semantically disambiguated.

Initial Detection Algorithm

The initial research and detection algorithm focused on manually defining a small set of Finite State Transducer (FST) [51] grammar rules used to perform the identification of grammatical syntax, application of the lexical rules such as the heuristic literary terms, and the detection of candidate metaphors from a small corpus. FSTs are essentially state machines that apply regular expression to the syntactic elements in a corpus. Each FST rule is responsible for evaluating the grammatical features associated with the part of speech tagged sentence to identify the primary or head noun and verb phrases in the sentence and then applied to the set of literary terms to the sentence. The General Architecture for Text Engineering (GATE) framework [29] was used to perform the basic parsing which includes splitting the given text into sentences, creating part of speech tags for each word in a sentence, splitting each sentence into noun and verb phrases, and identifying stem words. Within GATE the FST paradigm is implemented as Java Annotation Pattern Engine (JAPE) grammars.

A JAPE grammar consists of a set of phases, each phase consists of a set of pattern/action rules. A JAPE grammar is required to have two sides: Left and Right. The Left Hand Side (LHS) of the rule contains the identified annotation pattern that can contain regular expression operators. The Right Hand Side (RHS) outlines the action to be taken on the detected pattern and consists of statements for manipulating annotations. Labels are used to map annotations from the RHS to the LHS.

Literary Heuristics

In this scenario we were focused on the ability to identify metaphor candidates utilizing a set of heuristic literary terms as a means of detecting traditional metaphors. A traditional metaphor is one that follows the pattern X is (like) Y, where X and Y within a given text. These are the easiest metaphors to recognize and provide a means to programmatically create/establish a training set based on the SOURCE and TARGET domain associations and is the first step towards an unsupervised or semi-supervised approach to detecting metaphor candidates. Table 4-1 contains a partial list of literary heuristics applied to the given text.

Table 4-4. Literary heuristics

Literary Heuristic	Type
such	mixed
such as	simile
like	simile
is	mixed
is like	simile
is like a	simile
seemed	mixed
was	mixed
is a	simile
as a	simile
as if	mixed
as if it were	simile
which are	mixed
are	mixed
are but	mixed
to	mixed
to be	mixed

Each sentence containing one of the literary heuristics is selected as part of the first pass preprocessing required in identifying metaphor candidates. Applying the literary heuristics in Table 4-1 as a gazetteer, or concrete word list, in the NLP pipeline produces an annotation for each sentence and each word contained in the sentence

that corresponds to the LiteraryTerm. These annotations are created as metadata for every instance matched within the given text and are utilized within the JAPE grammar to detect a candidate metaphor. Looking at the sentence *"*The senate has become the battleground for health care lobbyists*,"* in order to detect the 'WAR' metaphor the literary term 'has become*'* would have to be added by hand to the list of literary terms.

The calculate confidence measure function returns a list of sentences from the given text, each sentence has a set of associated metadata that includes the parse information and if the sentence met any of the criteria to determine it was a literal sentence. Specifically, are the SOURCE and TARGET words semantically related as synonyms or hypernyms. The WordNet synsets are hierarchical collections of related word senses, here is where we define another heuristic to control the depth of the search through the chain assuming that at a depth of three there is a very small probability that the SOURCE and TARGET are related.

Confidence Algorithm

The confidence measure is calculated based on several factors that are used determine if a particular sentence has a probability of being a candidate metaphor. The confidence measure includes a calculation for weight of traversing the hypernym chain, with a maximum depth traversal set to 3 for this scenario. The probability that the sentence is a metaphor increases as the value of the hypernym weight increases. The inverse is also true if a hypernym is matched then there the probability increases that the sentence does not contain a metaphorical utterance. The confidence calculation for the proof of concept algorithm is quite simple; we take the sum of the semantic relationships identified for each of the sentence noun and verb phrases divided by the

total number of noun-verb pairs. Where ‘i’ is the different semantic relationships (e.g., hypernym) evaluated for each noun-verb pair.

$$Confidence_{POC} = \sum_{i} \frac{NounPhrase(i) + VerbPhrase(i)}{TotalNounVerbPairs(i)}$$

Robustness and limitations

The use of a heuristic literary term does several things; first it reduces the chance of false positives because it reduces the total number of sentences that are available to evaluate as candidate metaphors. Second, since the heuristic reduces the total number of sentences evaluated the end results are fewer metaphor candidates identified within the given text. Unfortunately this approach increases the number of false negatives or metaphor candidates that are missed. The JAPE grammar rules themselves are a limitation it is important to understand that the JAPES rules are essentially regular expressions, and they require the specific grammar types to be met otherwise the rule bypasses the potential candidate. Both of these approaches reduce the potential for false positives.

Extended Detection Algorithm

In the extended detection algorithm the FST grammars were completely removed from the processing pipeline. In this scenario we were focused on a more statistical approach to identify metaphor candidates still utilizing the set of heuristic literary terms (Table 4-1) as a means of detecting traditional metaphor candidates, those that follow the general pattern X is (like) Y, where X and Y are the SOURCE and TARGET concept domains within a given text. As with the initial version of the algorithm, the first step in the detection algorithm requires ingesting a corpus of documents. The algorithm then

performs the syntactic processing followed by semantic processing, and calculation of the confidence weight for each sentence.

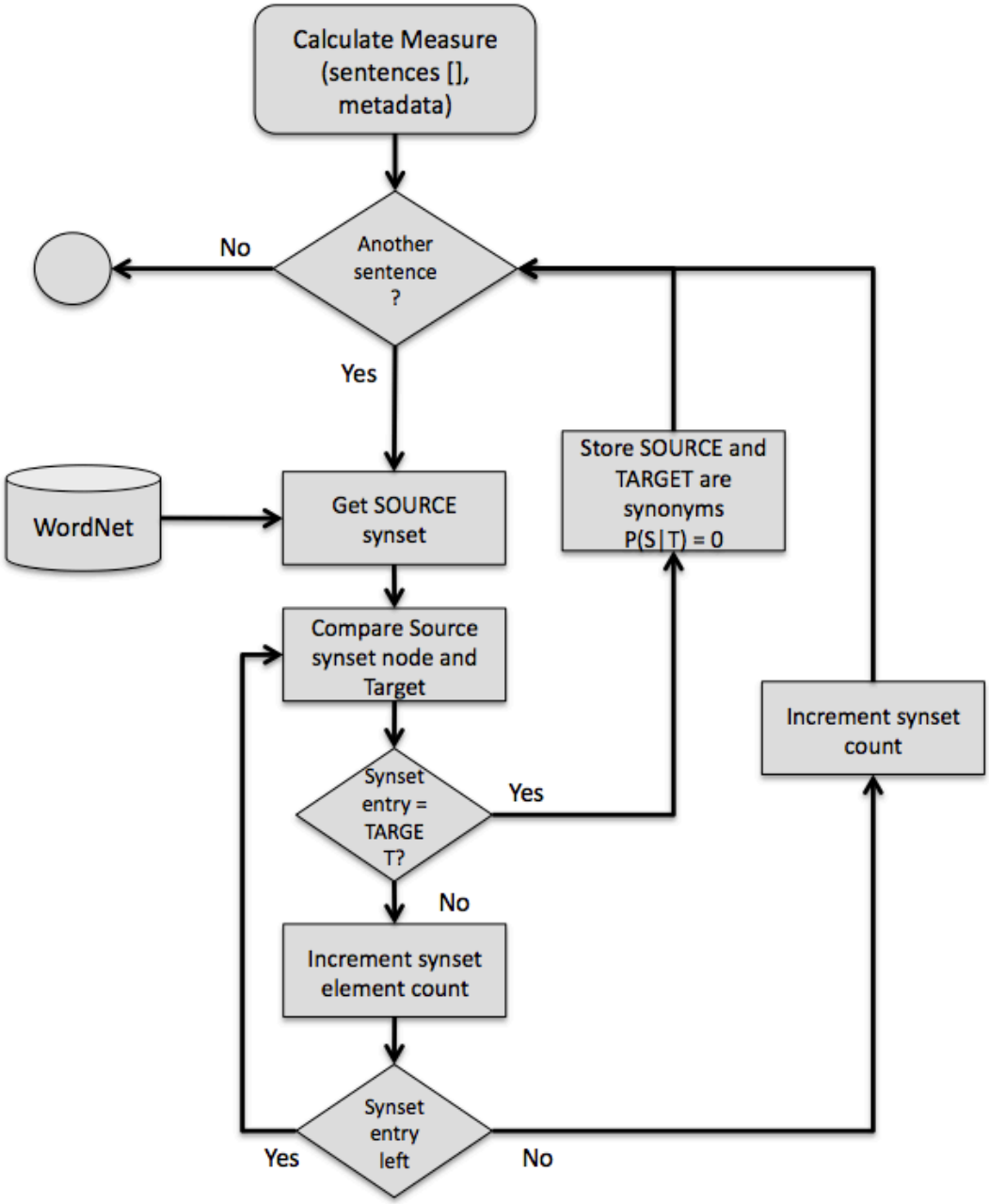

Figure 4-1. Calculate confidence measure

Syntactic Parsing

Syntactic parsing is the process of decomposing a given text into a hierarchy representing the grammatical components of the text. Syntactic parsing includes;

sentence splitting, word tokenizing, and part of speech tagging. Each parsed sentence can be represented as tree where the root node indicates the sentence bounded from first word to final punctuation.

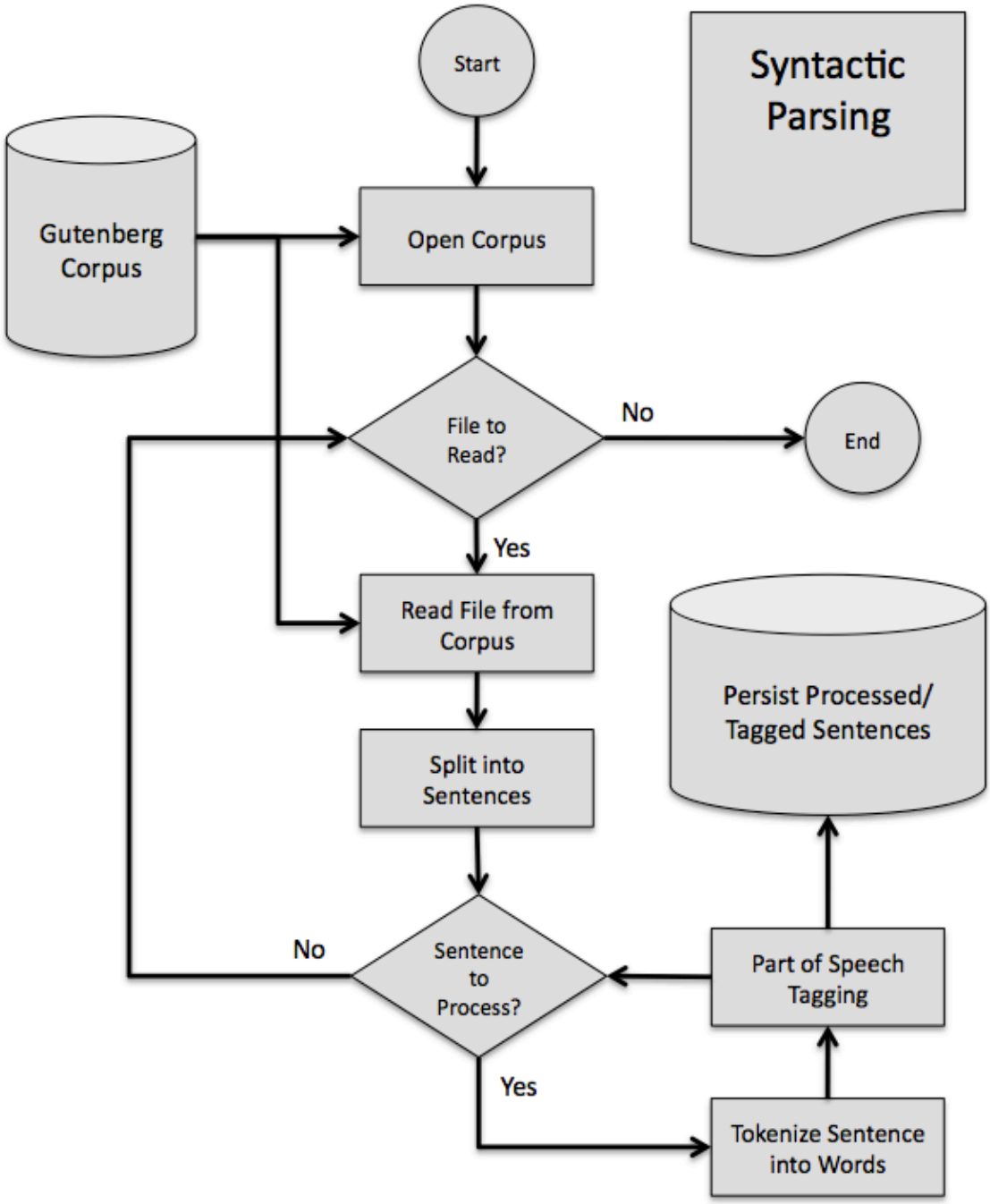

Figure 4-2. Syntactic Parsing Flow

Within the tree the terminal nodes are the individual words. The nodes between the terminal nodes and root node represent the part of speech tags using the Brill tagger [12]. The Brill tagger utilizes the part of speech tags defined by the Penn Treebank project [69]. The Penn Treebank is a bank of linguistic trees annotated for linguistic structure providing skeletal parse tree and provides one the accepted standard for part of speech tags where a noun = NN, verb = VB, and adjective = JJ (see Appendix A for a more extensive listing). Each tag is associated with the individual word and the identification of noun phrases (NP), verb phrases (VP), and prepositional phrases (PP). The individual words plus the part of speech tags provide a means searching for word senses and for establishing relationships between word pairs within a sentence, across a given text or an entire corpus.

> MARLEY was dead: to begin with. There is no doubt whatever about that. The register of his burial was signed by the clergyman, the clerk, the undertaker, and the chief mourner. Scrooge signed it: and Scrooge's name was good upon 'Change, for anything he chose to put his hand to. Old Marley was as dead as a door-nail.
>
> —Charles Dickens, *A Christmas Carol, pg 1*

Given the preceding paragraph we can perform the syntactic parsing. Figure 4-3 shows a typical syntax tree with the sentence containing a simple sentence structure composed of primary noun phrase and verb phrase. The head noun, typically the noun preceding the adjective clause, “Marley”, the verb, and adjective clause are easily identified. The phrase grouping and part of speech tagging provides the information for the grammatical relationship between each in the sentence and facilitates the identification noun-verb and noun-noun pairs.

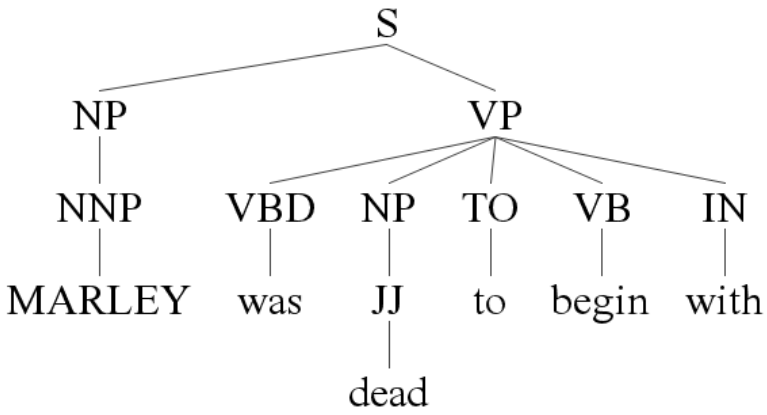

Figure 4-3. Syntax parse tree for “Marley was dead to begin with”

Another sentence structure includes the use of the existential “there”, this is an important exception to the use of the literary term plus the verb “is” from heuristic Table 4-4 applied to the metaphor form X is Y where X is not a noun. The syntax tree shown in Figure 4-4 contains an existential phrase where there is no true head noun. In this case the literary term would contribute negatively to the weighting calculation decreasing the probability that the sentence contains a metaphor candidate.

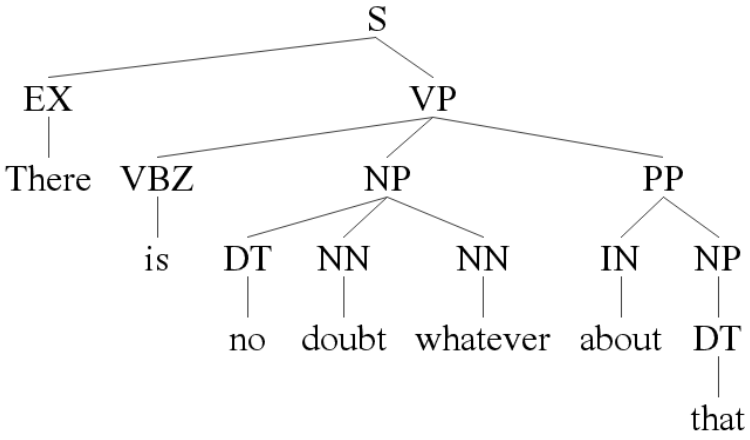

Figure 4-4. Syntax parse tree for “There is no doubt whatever about that”

Figures 4-5 and 4-6 show several more complex syntax trees containing multiple noun and verb phrases each.

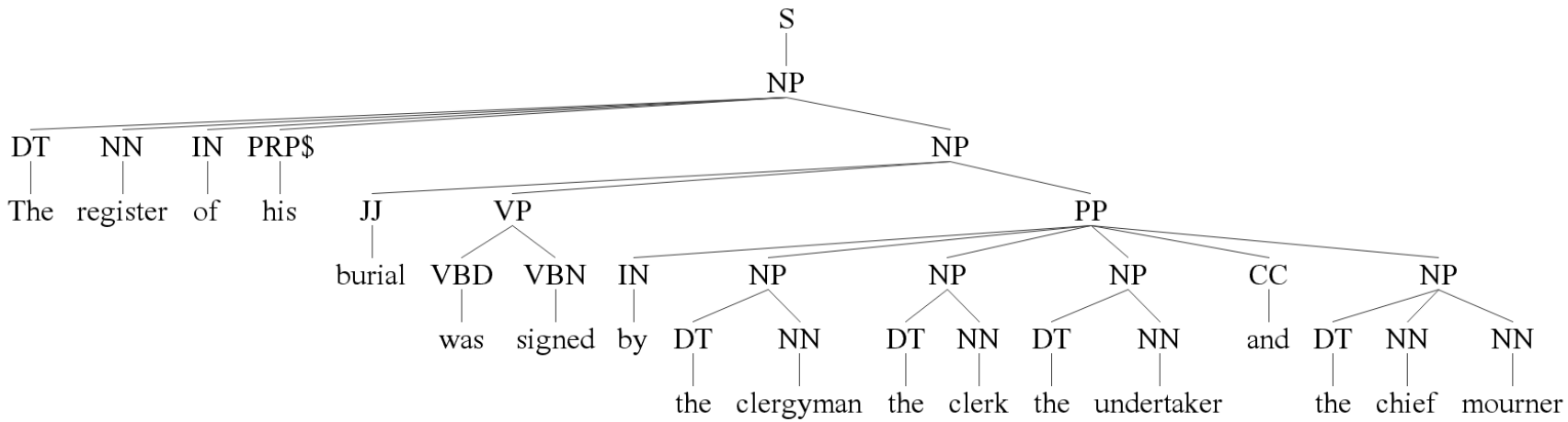

Figure 4-5. Syntax tree for “The register of his burial was signed by the clergyman, the clerk, the undertaker, and the chief mourner”

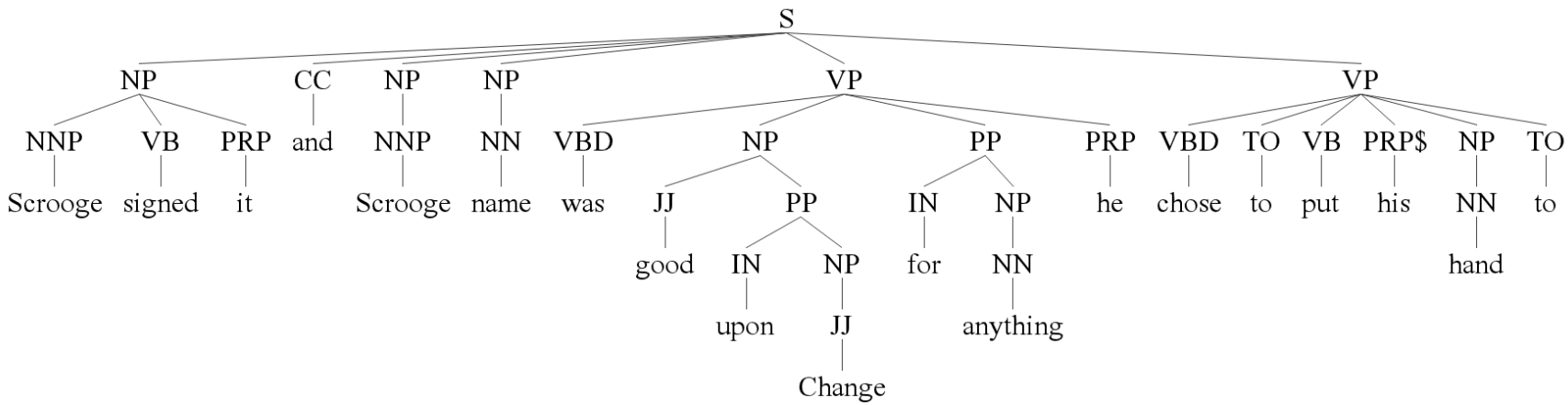

Figure 4-6. Syntax tree for “Scrooge signed it: and Scrooge's name was good upon 'Change, for anything he chose to put his hand to”

Figure 4-7 illustrates the final sentence in the opening paragraph. The head noun is “Marley”, the verb “was”, and the right most adjective of the two adjectives in the prepositional phrases (PP). The preceding verb is found in the list of heuristic literary terms, “was” from Table 4-4. In this case the literary term would contribute to the weighting calculation increasing the probability that the sentence contains a metaphor candidate.

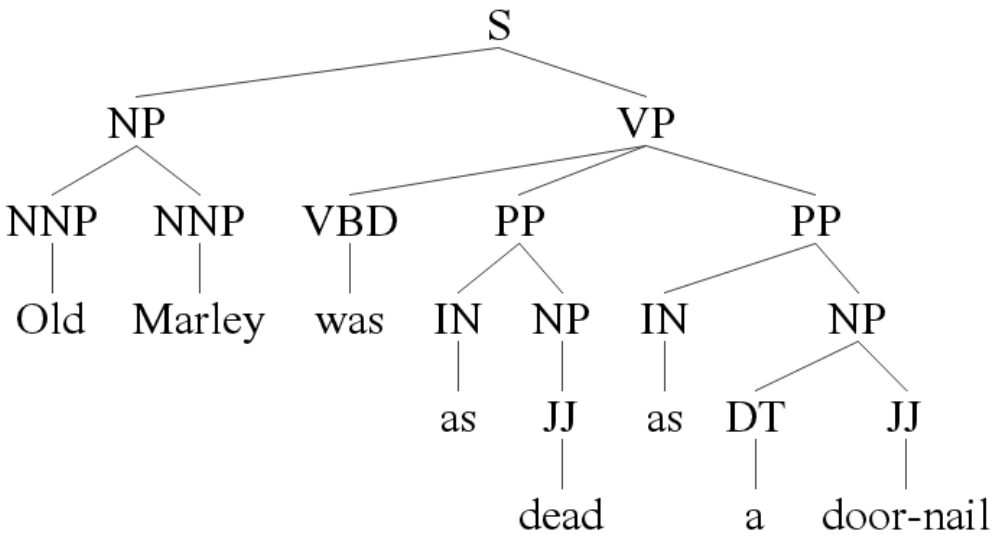

Figure 4-7. Syntax tree for “Old Marley was as dead as a door-nail”

Semantic Parsing and Word Sense

The syntactic parsing provides a significant amount of information. When combined with a simple set of guidelines or rules associated with SOURCE and TARGET domain concepts a human would most likely be able to identify metaphor candidates contained in the text. This is based on the ability of the human to draw on personal experiences and other a priori knowledge. However, a computer has no such experiences and needs to be provided a knowledge base containing sufficient information and semantic relationships to allow the inference of word senses and meaning.

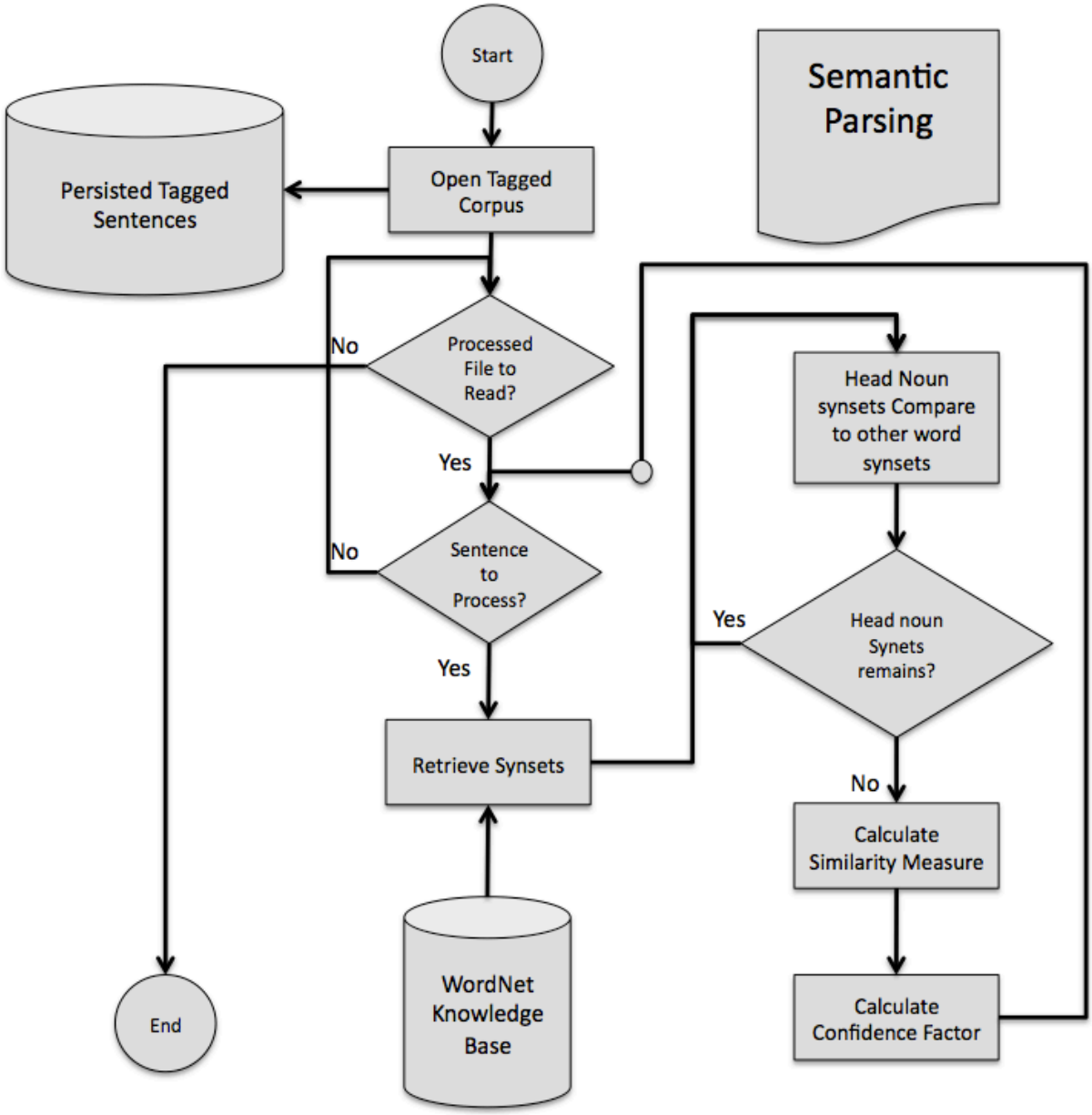

Figure 4-8. Semantic parsing flow

Given the previous syntactic parse trees and associated grammatical metadata from the example text representing the ingested corpus. The detection algorithm performs the semantic augmentation that includes identifying the lemma or root form of a word using the morphological analysis and word stemming, identifying the total

number of words, number of bigrams, word sense, and associated hypernym chains for each noun (NN), verb (VB), adjective (JJ), and adverb (RB) in each sentence.

Table 4-5. Example paragraph broken into sentences with words counts

ID	Sentence	Nouns	Verbs	Adj	Adverb	Total
1	Marley was dead to begin with	1	2	1	0	6
2	There is no doubt whatever about that					7
3	The register of his burial was signed by the clergyman, the clerk, the undertaker, and the chief mourner	6	2	1	0	18
4	Scrooge signed it: and Scrooge's name was good upon 'Change, for anything he chose to put his hand to	4	4	2	0	19
5	Old Marley was as dead as a door-nail	2	1	2	0	9

Identifying the semantic relationships for the individual words in a sentence begins with the set of synonyms associated with each word and the part of speech tag assigned during the syntactic processing. The synsets are key to evaluating the “Is-A” and “Part-Of” relationships between words. Each element of a synset constitutes a word sense, each word sense has a corresponding hypernym chain. Each word sense and associated hypernym chain is evaluated to determine the similarity of the word meanings. Indicators include the overall depth of the hypernym chains, if two hypernym chains intersect, and the depth at which the hypernym chains intersect. These measures contribute values between 0 and 1 to the overall calculation of the confidence factor used to identify if the sentence has a literal meaning or is a candidate metaphor.

Each synset is represented by label in the following format <lemma>.<pos tag>.<frequency>, where the <lemma> is the root word, <pos tag> is the part of speech tag, and <frequency> is the frequency of use for the given word sense within the training corpus. Table 4-6 shows the synset lists and the depth of the associated hypernym chain with the simple phrase “Marley was dead”. The synset list for each

word and associated hypernym chain provides the basis similarity comparison between the words associated with the SOURCE and TARGET concepts that could indicate a language metaphorical is contained in a sentence.

Table 4-6. Synset information for head noun phrase from example sentence 1

ID	Noun Synsets	Hypernym Depth	Verbs Synsets	Hypernym Depth	Adjective Synsets	Hypernym Depth
1	marley.n.01	1	be.v.01	1	all_in.s.01	1
			be.v.02	1	dead.s.04	1
			be.v.03	1	dead.s.05	1
			exist.v.01	1	dead.s.06	1
			be.v.05	1	dead.s.07	1
			equal.v.01	1	dead.s.08	1
			constitute.v.01	1	dead.s.09	1
			be.v.08	1	dead.s.10	1
			embody.v.02	10	dead.s.11	1
			be.v.10	3	dead.s.12	1
			be.v.11	1	dead.s.13	1
			be.v.12	3	dead.s.14	1
			cost.v.01	2	dead.s.15	1
					dead.s.16	1
					dead.s.17	1

Similarity measures

Similarity measure for two words is based on comparing features of the words, including word senses, part of speech, word tense, root word, and Is-A and Part-Of semantic relationships based on evaluating synonyms and hypernym/hyponym relationships. The semantic relationships are represented in a hierarchical fashion where the root node is the most general sense of a word and the leaf nodes are the most specific. The general comparison is a determination of the distance to the common root node in the hierarchy as well as the actual distance between the two words in the hierarchy. Each of the features compared are factored into the individual similarity measurement. For our purposes the hierarchy is the features contained in the WordNet

synsets. Since we are looking at an overall confidence factor we employ several similarity measurements to establish a better confidence in determining the salient characteristics between the SOURCE and TARGET domains that would represent a metaphor candidate including; shortest path, path similarity, depth of the hierarchy, least common ancestor, and information content for the least common ancestor.

Shortest Path Distance

The Shortest Path Distance measure calculates the shortest paths linking two synsets in WordNet if a path exists. For each synset, the distances for all ancestors, including the root, are calculated and then compared. The shortest path is indicated by the existence of a common ancestor node and the minimum number of nodes. It is possible that there is no common ancestor node between two synsets in this case the resulting weight is 0.

Path Similarity

The Path Similarity measure calculates a score denoting how similar two word senses are, based on the shortest path that connects the senses in the is-a (hypernym/hypnoym) synset linkage. The score is in the range 0 to 1, except in those cases where a path cannot be found. In the event there is no common ancestor, the value of 0 is returned indicating no similarity. The lack of a common ancestor typically occurs for verbs since there are numerous distinct verb taxonomies. If the synset is compared to itself then value of 1 is returned indicating an identical comparison.

Leacock-Chodorow Similarity

The Leacock-Chodorow Similarity measure calculates a score denoting how similar two word senses are, based on the shortest path that connects the senses in the

is-a (hypernym/hypnoym) synset linkage but also includes the maximum depth of the taxonomy in which the senses occur. The relationship is given as:

$$SimMeas_{LC} = -\log\left(\frac{p}{2*d}\right)$$

where *p* is the shortest path length and *d* is the hierarchy/taxonomy depth. The two words being compared must have the same part of speech resulting in a score > 0 only when both the SOURCE and TARGET concepts have the same part of speech (typically nouns).

Wu-Palmer Similarity

The Wu-Palmer Similarity measure calculates a score denoting how similar two word senses are, based on the depth of the two senses in the taxonomy and that of their Least Common Subsumer (most specific ancestor node). The LCS does not necessarily feature in the shortest path connecting the two senses, as it is by definition the common ancestor deepest in the taxonomy, not closest to the two senses. Typically, however, it will exist as a feature in the shortest path. In the case where multiple candidates for the LCS exist the longest shortest path to the root node is selected. Where the LCS has multiple paths to the root, the longer path is used for the purposes of the calculation. The similarity measurement is represented by:

$$SimMeas_{WP} = \left(\frac{2*d}{(p1+d)+(p2+d)}\right)$$

where *d* is the depth in the taxonomy/hierarchy, *p1* is the shortest path the SOURCE, and *p2* is the shortest path for the TARGET. If the subsumer's part of speech is not that of a noun, then 1 is added to the depth, this is done since the expectation is that noun will have an ancestor root.

Resnik Similarity

The Resnik Similarity measure calculates a score denoting how similar two word senses are, based on the Information Content (IC) of the Least Common Subsumer (most specific ancestor node). Resnik similarity requires the information content of the least common subsumer that has the highest information content value. If two nodes in the taxonomy/hierarchy have no explicit common subsumer, the assumption is that they share an artificial root node that is the hypernym of all explicit roots.

Jiang-Conrath Similarity

The Jiang-Conrath Similarity measure calculates a score denoting how similar two word senses are, based on the Information Content (IC) of the Least Common Subsumer (most specific ancestor node) and that of the two input words and synsets features. The relationship is given by the equation:

$$SimMeas_{JC} = \frac{1}{(IC(s1) + IC(s2) - 2 * IC(lcs))}$$

where *s1* is the SOURCE word synset, *s2* is the TARGET word synset, and *lcs* is the most specific ancestor.

Lin Similarity

The Lin Similarity measure calculates a score denoting how similar two word senses are, based on the Information Content (IC) of the Least Common Subsumer (most specific ancestor node) and that of the two input synsets defined for the SOURCE and TARGET words. The relationship is given by the equation:

$$SimMeas_{Lin} = \frac{2 * IC(lcs)}{(IC(s1) + IC(s2)}$$

where *s1* is the SOURCE word synset, *s2* is the TARGET word synset, and *lcs* is the most specific ancestor.

Literary Heuristics

Each sentence containing one of the literary heuristics is selected as part of the first pass preprocessing required in identifying metaphor candidates. Applying the literary heuristics in Table 4-1 provides an additional metaphor utterance measurement included in the confidence calculation. If one of the terms is contained in the sentence then a 1 is returned otherwise a 0. Looking at the sentence *"*The senate has become the battleground for health care lobbyists*,"* in order to detect the 'WAR' metaphor the literary term 'has become*'* would have to be added by hand to the list of literary terms and would return the value of 1.

Confidence Algorithm

The calculate confidence measure function returns a list of sentences from the given text, each sentence has a set of associated metadata that includes the parse information and if the sentence met any of the criteria to determine it was a literal sentence. Specifically are the SOURCE and TARGET words synonyms, are the SOURCE and TARGET hypernyms and to what degree (how closely related).
The WordNet synsets are hierarchical collections of related word senses, here is where we define another heuristic to control the depth of the search through the chain assuming that at a depth of three there is a very small probability that the SOURCE and TARGET are related.

$$Confidence_{Full} = \sum_{i} \frac{SimMeas(i) + LiteraryHeuristic + Selectional \Pr_R(P)}{NumBigrams}$$

The confidence measure is calculated based on the summation of the similarity measurements and the literary heuristic this returns a weighted value that is used to indicate if a sentence is a candidate metaphor. Where '*i*' is one of the similarity

measurements, SimMeas(*i*) is the largest value of all the similarity comparison for '*i*', and NumBigrams is the number of bigrams, word pairs, in the sentence.

Robustness and limitations

The use of a similarity measurement provides a mechanism for identifying semantic relationships between the SOURCE and TARGET words and associated synsets specifically associated with the ability to detect literal and metaphorical language. The Selectional Preferences factor provides a discriminator indicating the head noun and head verb are commonly used together and acts as a negative weighting factor. The heuristic literary terms provide an additional weighting factor indicating a metaphor candidate, however, it no longer acts as an explicit filter eliminating sentences that do not contain one of the literary terms. The similarity measurements are limited to the content of the machine-readable dictionary providing the lexical semantic information for the word senses. Lastly, the different factors included in the confidence calculation can be used to cluster the metaphor candidates based on different combinations of the semantic features.

CHAPTER 5
ANALYSIS AND RESULTS

Objectives

To test the feasibility of a detection algorithm effectively utilizing a literary term heuristic in conjunction with the semantic relationships identified for the SOURCE and TARGET concept domains of a given sentence we performed several experiments. The first experiment focused on the viability of using a set of literary terms (see Table 4-1) that are traditionally associated with linguistic metaphors of the form X is (like) Y. The second experiment focused on the detection of semantic relationships and calculating a value indicating similarity of the SOURCE and TARGET concepts using the WordNet synsets to evaluate the hypernym and hyponym relationships. The final experiment focused on defining a confidence factor based on the literary term heuristic, the semantic relationship similarities, and linguistic statistics (e.g., number of bi-grams).

Both the syntactic and semantic processing employ a brute force method performing an exhaustive comparative search through the semantic relationships of the individual words. The literary term heuristic is not intended to accelerate the semantic relationship search only to add a linguistic cue weight to the overall confidence calculation. We also make the assumption that since WordNet is based on a hierarchy that most if not all hypernym/hyponym chains converge on a common root (e.g., entity) that is too general to provide a useful weight factor for the confidence calculation. Based on our observations of the hypernym chains we selected an initial bounding value of 3. For all of our experiments we make the assumption that simile is a subset of metaphorical language and therefore we do not delineate between them.

Accuracy Evaluation

In natural language processing, F-measure is the accepted measure for determining the accuracy of results developed by the National Institute for Standards and Technology (NIST) to measure effectiveness of information retrieval systems [66]. The formula is similar to pattern recognition measurement utilizing the metrics of ‘precision’ and ‘recall’ to compute the F-measure. Where ‘precision’ is the ratio of correct instances of the item being searched for to the set of perceived instances found by the algorithm providing a measure of exactness and ‘recall’ is the ratio of the number of correct instances of the item being searched for to the set of actual relevant instances providing a measure of completeness.

precision = (relevant # of elements retrieved) /(total # of elements retrieved)

recall = (relevant # of elements retrieved) / (total # of relevant elements)

F-measure is the weighted average or harmonic mean of the ‘precision’ and ‘recall’.

$$F = 2 * \left(\frac{precision * recall}{precision + recall} \right)$$

These simple metrics are used to compute the fraction of instances for which the correct result is returned. As the score approaches 1 the accuracy is better, the accuracy is less as the value approaches 0.

As part of the analysis we created baseline set of texts also known as a gold data set. The gold data set is a set of texts that have been manually reviewed sentence by sentence, preferably by a linguistics expert or other specific domain expert. The reviewer identifies metaphor instances within the text, including the explicit sentence or phrase and the location in the text. For the experiments associated with this research

the gold data set was created by an individual with no a priori knowledge of the capabilities of the prototype supporting this research.

A corpus of 16 texts was selected at random from the Project Gutenberg corpus. Each text was reviewed and the set of metaphors and location in the text were manually identified and persisted for use with the final F-measure calculation. The objective was to identify all the possible metaphors independent of the prototype development so that the F-measure calculation could be performed and the result would accurately indicate the capabilities of the metaphor detection algorithm.

Table 5-1. Metaphors identified in the gold data set

Text Name	Metaphor Count
Anderson Fairy Tales	17
Around the World in eighty Days	70
Art of War	39
At the Earth's Core	19
Alice's Adventures in Wonderland	17
Through the Looking Glass	25
Red Badge of Courage	98
The Legend of Sleepy Hollow	26
Grimm's Fairy Tales	25
House of the Seven Gables	8
Paradise Lost	120
Peter Pan	14
The Prince and the Pauper	6
The Return of Sherlock Holmes	11
Treasure Island	38
What is Man and Other Stories	8

Experiment 1

This experiment focused on the viability of using the set of literary terms defined in Table 4-1. These literary terms are linguistic cues used by humans when searching for metaphorical language of the form X is (like) Y. The application of these cues acts as a filter in that sentences that do not contain a literary term are excluded from further

evaluation in the algorithm. The expected result is a higher number of misses and fewer false positives. We make this assertion based on our research and the knowledge that literal metaphorical language is not constrained to form X is (like) Y and verbs other than ‘be verbs’ in metaphorical utterances. For this experiment we used a corpus containing a single document, “Alice’s Adventures in Wonderland”.

Environment

The first experiment was performed with an algorithm that included Finite State Transducers (FST) grammar rules, Java Annotation Pattern Engine (JAPE), developed using Java and the General Architecture for Text Engineering (GATE) Integrated Development Environment (IDE). The natural language processing pipeline and association plugins (i.e., ANNIE) were orchestrated, loaded, and executed within the GATE IDE. The test runs were performed on a MacBook Pro with a 2.66 GHz Intel i7 quad processor, 8 GB 1067 MHz DDR3 Memory, using Mac OS X 10.6.7 on a 64 bit kernel.

Results and Observations

Table 5-2 contains the metaphors that were detected using the using the literary term heuristic as a binary filter and hypernym chaining with a bounding limit of 3. The table contains two pieces of data; 1) the text of the sentence, 2) part of speech mapping for each sentence. The part of speech mappings are in indicator of the grammatical complexity that can be associated with sentences that have the form X is (like) Y providing evidence that both a hand coded grammar rule and rote learning approach to applying grammars rules would result in a significantly large set of patterns and is not feasible.

Table 5-2. Metaphor candidates from Alice in Wonderland detected using the heuristic literary terms and the hypernym chaining

Metaphors Found	Associated Part of Speech
away went Alice like the wind	<RB><VBD> <DT><NN>
I could shut up like a telescope!	<PRP><MD><VB><IN>root=like<DT><NN>
shutting people up like telescopes	<VBG><NNS><RB><IN>root=like<NNS>
I went to the Classics master, though. He was an old crab, HE was.'	<NNP><MD><VB><PRP><VBG><IN>root = as<IN><PRP$><NN><MD><VB>
going back to yesterday,	<RB><PRP><VBP>root=be<RB><JJ>
explanations take such a dreadful time.'	<PRP><VBD>root=go<TO><DT><NNS><NN> . <RB><punctuation><PRP><VBD>root = be<DT><JJ>
he is gay as a lark,	<VBG>root=go<RB><literaryTerm & TO><NN>
I don't believe there's an atom of meaning in it.'	<NNS><VPB>root=take<PDT>root=such<DT><JJ><NN>
There was a dead silence	<PRP><literaryTerm><VBD>root = be<JJ><literaryterm><IN>root = as<DT><NN>
something comes at me like a Jack-in-the-box,	<NN><VBZ><IN><PRP><IN>root=like<DT><NNP>
and up I goes like a sky-rocket!'	<CC><RB><PRP><VBZ>root=go<IN>root=like<DT><JJ>
an immense length of neck, which seemed to rise like a stalk out of a sea of green leaves that lay far below her	<DT><JJ><NN><IN><NN><punct><WDT><VBD><TO><VB><IN>root=like<DT><NN><IN><IN><DT><NN><IN><JJ><NNS><WDT><VBP><RB><IN><PRP>
her neck would bend about easily in any direction, like a serpent	<PRP$><NN><MD><VB><IN>root=about<RB><IN><DT><NN><punctiation><IN>root=like<DT><NN>
with a round face, and	<JJ><NNS><IN>root=like<DT><NN>
large eyes like a frog	<RB><IN>root=like<DT><JJ>
'just like a star-fish,'	<DT><JJ><JJ><NN><VBD>root = be<VBG><IN>root=like<DT><JJ>
The poor little thing was snorting like a steam-engine	<NNP><MD><VB><PRP><VBG><IN>root = as<IN><PRP$><NN><MD><VB>
Alice could hear him sighing as if his heart would break	<RB><PRP><VBP>root=be<RB><JJ>
really you are very dull	<DT><JJ><JJ><NN><VBD>root = be<VBG><IN>root=like<DT><JJ>

Evaluation

In order to evaluate the effectiveness of the heuristic literary term a comparison of the set of metaphors found in the text by the algorithm and those defined in the gold data set needs to be performed. Table 5-3 contains the set of metaphors that were identified by hand.

Table 5-3. Metaphor candidates identified manually for the gold data set

Metaphors Found
The rabbit-hole went straight on like a tunnel for some way…
There are no mice in the air, I'm afraid, but you might catch a bat, and that's very much like a mouse, you know
… away went Alice like the wind
"Oh, how I wish I could shut up like a telescope!
… book of rules for shutting up people like telescopes…
What a curious feeling!" said Alice; 'I must be shutting up like a telescope.'
…in my going out altogether, like a candle.
now I'm opening out like the largest telescope that ever was!
…an immense length of neck, which seemed to rise like a stalk out of a sea of green leaves…
…her neck would bend about easily in any direction, like a serpent."
There could be no doubt that it had a VERY turn-up nose, much more like a snout than a real nose.
…because the chimneys were shaped like ears and the roof was thatched with fur.
'Why is a raven like a writing-desk?'
…and there stood the Queen in front of them, with her arms folded, frowning like a thunderstorm.
…he is gay as a lark…
'…or else you'd have signed your name like an honest man.'
"'Do I look like it?'"

Ideally this would be a completely comprehensive list of all metaphors contained in the given text. However, analysis shows that both the human developing the gold data set for evaluation and the detection algorithm missed numerous instances of metaphorical language, which is not a surprising result. Our research has shown that detection of metaphorical language is difficult even for humans. The set of metaphors contained in the gold data set demonstrated a higher accuracy with respect to falsely identifying a

metaphor candidate. The algorithm missed more than the human that developed the gold data set. However, the comparative analysis of the actual metaphor instances in the gold data set and those detected by the algorithm indicated that the human also missed a significantly high number (see Table 5-4).

To calculate the F-measure we must assume that the sum of both totals less than those in common and the 3 false positives equates to the total number that exists in the text. The F-measure results indicate that the literary term heuristic is an effective discriminator for detecting metaphorical utterances.

Table 5-4. Comparison of metaphors detected

Metaphors	Total	Common	Misses	False Positives
Human	17	5	10	0
Heuristic and Hypernym	18	5	12	3

F-measure

- Precision: (17/17)*100 = 100 %
- Recall: (17/30)*100 = 56 %
- F-measure Algorithm: (2 * (.56/1.56))*100 = 71.7 %

Experiment 2

This experiment focused on the viability of using semantic relationships detected within a sentence and calculating a value indicating similarity of the SOURCE and TARGET concepts. The semantic relationships are evaluated using the WordNet synsets and the associated hypernym and hyponym relationships. The algorithm in this experiment extended the use of the literary terms defined in Table 4-1 from the filter approach in experiment 1 to an individual weighting factor. This approach results in the literary term heuristic not explicitly excluding sentences from further evaluation. The expected result is a fewer number of misses and the potential increase in false

positives. We make this assertion based on our research and previous experimental results. For this experiment we used a corpus consisting of individual sentences utilized in other metaphor detection systems.

Environment

The second experiment was performed with an extended version of the algorithm from experiment 1. The algorithm was developed using Python and the Natural Language Toolkit (NLTK) for the processing pipeline and the WordNet interface. The test runs were performed on a MacBook Pro with a 2.66 GHz Intel i7 quad processor, 8 GB 1067 MHz DDR3 Memory, using Mac OS X 10.6.7 on a 64 bit kernel.

Results and Observations

Table 5-5 contains the list of sample sentences along with weight values calculated from the literary term heuristic, the similarity measurements, and resulting confidence factor. For the confidence factor we consider any value greater than .55 to be a candidate metaphor. We observed a number of unexpected results with this set of samples. Sentences 7 and 9 have no similarity measurements. This is due to SOURCE and TARGET domain part of speech being different where one is a noun and the other it a verb resulting in a confidence factor based purely on the heuristic value and the number of bi-grams in the sentences. Sentence 7 results in a missed metaphor and sentence 9 with a very low confidence factor also resulting in a missed metaphor. Sentences 1, 3, and 4 have a lower than expected confidence factor, all of these sentences include pronouns, indicating there is an additional linguistic feature that needs to be resolved, specifically what does the pronoun mean. One potential assumption is a pronoun indicates a human. Therefore when a pronoun or proper noun is detected the synset for human will be used for similarity the comparison.

Table 5-5. Confidence factor head noun and verb from a sample set of sentences

ID	Sentence	Literary Term	Shortest Path	Path Similarity	Leacock-Chodorow Similarity	Wu-Palmer Similarity	Resnik Similarity	Jiang Conrath Similarity	Lin Similarity	Confidence Factor
1	He is a shark in a pool of minnows	1	0	0.086	0.09	0.15	0	0.12	0	.18
2	My lawyer is a shark	1	0	0.07	1.07	0.15	2.15	0.05	0.18	.72
3	He is a snake	1	0	0.07	0.08	0.15	0	0.05	0	.45
4	She is a shrew	1	0	0.05	0.07	0.15	0	0.05	0	.44
5	The defense killed the quarterback	0	0	0.06	0.92	0.125	0	0.04	0	.28
6	My car drinks gasoline	0	0	0.07	1.15	0.266	0.77	0.06	0.06	.79
7	How do I kill a process	0	0	0	0	0	0	0	0	0
8	How do I get into lisp	1	0	0.33	1.64	0.25	1.7	0.079	0	.99
9	The sky is falling	1	0	0	0	0	0	0	0	.33

Evaluation

All of the sentences in Table 5-5 are metaphorical utterances leaving a total set of 9 metaphors. The algorithm calculated confidence factors that three of the nine sentences were candidate metaphors, four of the sentences had greater than .28 value but less than a .5 value, and one complete miss. Based on our observations three of the four misses by the algorithm are easily rectified with the inclusion of a noun heuristic to handle pronominal co-reference and the use of proper nouns, specifically individual names.

F-measure no Pronoun Rule

- Precision: (3/8)*100 = 37 %
- Recall: (3/9)*100 = 33 %
- F-measure Algorithm: (2 * (.124/.70))*100 = 35.4 %

F-measure with Pronoun Rule

- Precision: (6/8)*100 = 75 %
- Recall: (6/9)*100 = 66.7 %
- F-measure Algorithm: (2 * (.5/1.417))*100 = 70.5 %

Experiment 3

This experiment focused on the overall ability of the algorithm to detect metaphorical language in a general corpus using the literary term heuristic, semantic relationships similarity weights, and a pronoun found weighting factor. The literary term heuristic is used as a weighting factor and not as an explicit filter to exclude or include sentences. The semantic relationships are evaluated using the WordNet synsets and the associated hypernym and hyponym relationships. The expected result is a fewer number misses and no increase in false positives. We make this assertion based on our research and previous experimental results. For this experiment we used a corpus consisting of Project Gutenberg texts listed in Table 5-1.

Environment

The third experiment was performed with the version of the algorithm from experiment 2 extended with a pronoun weight factor. The development was done using Python and the Natural Language Toolkit (NLTK) for the processing pipeline and the WordNet interface. The test runs were performed on a Mac Pro with a 2.88 GHz Intel Xeon quad processor, 18 GB 800 MHz DDR2 Memory, using Mac OS X 10.6.7 on a 64 bit kernel.

Results and Observations

The processing of the general corpus using the exhaustive comparison of the WordNet synsets presented a significant performance challenge. By default the Python shell environment was only running on a single core and was taking between five and fifteen minutes to process an individual sentence. Since there was no interdependency between the documents within the corpus, we ran six documents in parallel across the available eight CPU cores. For the confidence factor we still consider any value greater than .55 to be a candidate metaphor.

Evaluation

Each of the documents in the corpus was processed using the detection algorithm and the results collected. The collected results were compared to the set of metaphorical utterances found by hand. The evaluation is based on the total number of metaphor candidates identified, based on the results from experiment 1 we also need to account for candidate instances that were found by the human and not by the algorithm and vice versa.

Table 5-6. Comparison of metaphors detected from general corpus

Text Name	Hand Found	Hand Misses	Hand False Pos	Algorithm Found	Algorithm Misses	Algorithm False Pos	Common	Total
Anderson Fairy Tales	57	1	0	36	21	0	20	58
Around the World in eighty Days	70	0	0	64	6	4	64	70
Art of War	39	0	0	19	20	2	19	39
At the Earth's Core	19	0	1	18	0	0	18	18
Alice's Adventures in Wonderland	17	10	0	22	8	3	9	30
Through the Looking Glass	25	0	0	20	5	0	20	25
Red Badge of Courage	98	0	0	84	14	2	84	98
The Legend of Sleepy Hollow	26	0	0	22	4	0	22	26
Grimm's Fairy Tales	26	2	0	13	13	4	13	28
House of the Seven Gables	8	5	0	10	3	2	5	13
Paradise Lost	120	0	0	103	17	10	103	120
Peter Pan	14		0					
The Prince and the Pauper	6	18	0	18	0	5	0	18
The Return of Sherlock Holmes	11	7	2	15	1	2	9	16
Treasure Island	38	0	0	31	7	3	31	38
What is Man and Other Stories	8	7	0	13	2	1	6	15

We also removed any candidates from the gold data set that might be considered a false positive. However due to the length of the texts and number of sentences in each text it is outside the scope of this research to insure that there are no missed metaphors in the gold data set. Instead we will calculate the expected total from the metaphors found by hand plus the metaphor candidates from the algorithm, less the candidates from the intersection of the two sets.

F-measure

Table 5-7. Experiment 3 F-Measure results

ID	Text Name	Precision	Recall	F-Measure
1	Anderson Fairy Tales	100.00%	62.07%	76.60%
2	Around the World in eighty Days	93.75%	85.71%	89.55%
3	Art of War	89.47%	43.59%	58.62%
4	At the Earth's Core	100.00%	100.00%	100.00%
5	Alice's Adventures in Wonderland	86.36%	63.33%	73.08%
6	Through the Looking Glass	100.00%	80.00%	88.89%
7	Red Badge of Courage	97.62%	83.67%	90.11%
8	The Legend of Sleepy Hollow	100.00%	84.62%	91.67%
9	Grimm's Fairy Tales	69.23%	32.14%	43.90%
10	House of the Seven Gables	80.00%	61.54%	69.57%
11	Paradise Lost	90.29%	77.50%	83.41%
12	Peter Pan	100.00%	71.43%	83.33%
13	The Prince and the Pauper	72.22%	72.22%	72.22%
14	The Return of Sherlock Holmes	86.67%	81.25%	83.87%
15	Treasure Island	90.32%	73.68%	81.16%
16	What is Man and Other Stories	92.31%	80.00%	85.71%

A number of the metaphors identified were complex containing more than one instance of metaphorical language. Ideally the gold data set would have been created as a double blind test, where two sets are independently created by hand and the intersection used as the baseline set of metaphors. One of the items that became apparent during the analysis is that the there is an inconsistency in the manual metaphor identification between the individual texts. The number of metaphors detected by the algorithm and missed during the manual annotation indicates that the manual

effort was either cursory or focused on a specific metaphor pattern and not general metaphoric language.

CHAPTER 6
CONCLUSIONS

Metaphors are more than just a decorative linguistic feature they are pervasive in all forms of written communication. They provide a symbolic representation to concepts that are difficult to describe as well as insights into the emotions, thoughts, and ideology for individual, social groups, and cultures. This makes the detection and classification of metaphors of interest to a broad range of disciplines from areas in psychology to intelligence analyst.

We argue that while the typical form of a metaphor is described as; X is (like) Y or X = Y and that metaphor detection is not a binary evaluation. We also argue that metaphor detection requires the consideration of a number of factors including empirical knowledge that includes semantic relationships and context. Without these factors there is a limit to the linguistic metaphors that can be detected. The related work has focused on small domains using manually defined rules that evaluated each sentence to determine if it is violating a literal meaning.

In this research we present an algorithm that employs a literary term heuristics and several boundary conditions rules aimed at the detection of parts of speech that include accounting for the use of existential language and pronouns. Our algorithm does not treat determination of a sentence as literal or metaphorical, instead we identify salient characteristics that may indicate metaphorical language. Since each of the heuristic rules contributes to the confidence calculation this approach allows for new feature weights to be easily added to the confidence calculation.

The experimental results associated with the calculated confidence factor provide a reasonable indication that a given sentence contains metaphorical language. One

item that we discovered in our research of similar systems is that a measurement of accuracy is not provided. In fact Baumer [8] identifies the need for measuring the accuracy, and that it is defined as follow on work.

In this research we present a comparison of the detection algorithm results to a set of manually evaluated texts and the set of metaphorical statements identified, or gold data set. The F-measure measurement is based the idea of precision and recall or more specifically the ratio of how many metaphors were found that are correct, to how many metaphors actually exist in the given text. This is an extremely laborious and requires a significant level of expertise for the person performing the evaluation. For our research purposes we engaged an undergraduate student to evaluate the texts listed in Table 5-1 and identify the set of metaphors contained in the gold data set.

Another significant observation was the lack of metaphors in certain text. Given the statistic that people use metaphors once every six minutes or every 10 to 25 words, the number of metaphors in the gold data set and those identified by the algorithm are significantly less than number expected. Upon further investigation we discovered that the children's stories contained fewer numbers of metaphors and that fictional and stories about conflict and war such as "The Red Badge of Courage" contain significantly more metaphors. This appears to be intentional by the authors based on the target audience since a child's cognitive abilities are not fully developed.

We present the resulting F-measure comparison of the metaphor detection results from experiment 3 results with those of the gold data set (see Table 5-5). The measurement is intended to illustrate the accuracy of the metaphor detection. For an F-measure calculation to be accurate all of the metaphors have to have been identified in

the given text. Unfortunately, our experiment has shown that the manually identified metaphors are significantly less than what is actually contained in the text. While not ideal, this does serve to support our assertion that metaphor detection is hard even for humans, and that metaphor detection is not a binary question of true or false.

Contributions

Our goal in this research was to contribute to the knowledge in the area of Artificial Intelligence as designated by ACM's Computing Classification System. The contribution to Artificial Intelligence was two fold – the first contribution was in the method used to detect metaphors based on simple heuristics based on the X is (like) Y, statistical methods for similarity, unsupervised learning/clustering techniques to identify salient characteristics between the individual words in a given sentence utilizing machine-readable dictionaries containing lexical and semantic information. The second contribution was the more specific area of Computational Linguistics where a confidence factor was used to indicate metaphor candidate since detection is not a binary (true/false).

Future Work

Literary Analysis

We presented several use cases related to literary analysis including metaphor criticism, cultural understanding (intelligence analyst), and conflict resolution. The automated detection of metaphors could prove useful in the analysis of current and classical literature, cultural websites and blogs, social networks, emails, and chats.

The use of metaphors is not bounded by just genre but also temporally. Metaphors in use one hundred years ago may not be in use today, or they may be everyday terms. Metaphors used by teenagers are not necessarily the same as those used by adults.

The automated detection of metaphoric language could facilitate the ability to compare a large collection of work collected over many years and provide an understanding as to how people and cultures have changed over that time period.

Improvements

As described, the detection algorithm uses a collection of semantic relationships to provide a weighted measure to for the salient characteristics between the SOURCE and TARGET concept domains. We use the set of WordNet synsets [75] from both the SOURCE and TARGET domains along with information about their convergence and depth of the hypernym chain. We currently employ a non hierarchical to grouping of sentences based on the semantic relationships, specifically the literary heuristic and the similarity measurements.

These same sets of semantic relationships provide a number of characteristics that can be used in a number of clustering techniques. Algorithms such k Nearest Neighbor (kNN) are agglomerative clustering techniques, meaning that each data point starts as an individual cluster containing only that data point and that similar clusters are progressively combined to form larger clusters. Since the corpus is treated as one large set of sentences divisive clustering could be used to progressively divide the sentences into smaller clusters based on the confidence factors. Both agglomerative and divisive clustering determine the stopping point by create implied hierarchies based on the order that clusters are formed.

Additional Weights

The exhaustive search paradigm employed to evaluate the synsets and the corresponding semantic relationships presents a number of opportunities. The inclusion of a ontology such as the Suggested Upper Merged Ontology (SUMO) could provide a

means for relating the synsets and obfuscate the need to perform an exhaustive search. There is an existing mapping from WordNet to SUMO and this technique is being employed in the area of mapping literal metaphors to conceptual metaphors [3]. Other significant performance improvements can be made by changing the algorithm to execute in a distributed environment with multiple CPUs.

Another option is the incorporation of metaphor databases such as Mind is a Metaphor [82] or Metabank [71] as a training set for creating clusters for syntactic and semantic relationships of a large number of known metaphors. This approach could provide a performance improvement as well as an overall improvement in accuracy. Unfortunately, these online resources are not currently structured for this type of programmatic evaluation.

Publishing Gold Data Set

We identified a lack of existing data sets available for comparing and evaluating the results of automated metaphor detection. Establishing a gold data set is laborious at best and requires many hours of manually reading and annotating texts. One objective for future work would be to correct the deficiencies associated with the metadata indicating the metaphor and location in the gold data set used in this research. The gold data set could then be published to a freely available a repository of reference data such as National Institute of Standards and Technology (NIST) Standard Reference Data (SRD)[15]. Another item that would make the gold data set and the experimental results more useful would be the inclusion of negative examples. The negative examples would serve to provide a check of how well any detection algorithm does

[15] NIST Standard Data Repository: http://www.nist.gov/srd/

against a specific set of sentences that contain instances of a heuristic literary term as well as noun – verb pairs that are not common in the corpus but are viable in a literal sentence.

APPENDIX A
PART-OF-SPEECH TAGS FROM THE PENN TREEBANK PROJECT

Table A-1. Part-of-speech tags

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	To
UH	Interjection
VN	Verb, base form
VBD	Verb, past tense
VBG	VBG Verb, gerund or present participle
VBN	VBN Verb, past participle
VBP	VBP Verb, non-3rd person singular present
VBZ	VBZ Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

LIST OF REFERENCES

[1] Aristotle, “Poetics,” in *Ars Poetica*, Translated by S.H. Butcher, [Online]. Available: http://www.davemckay.co.uk/philosophy/aristotle/aristotle.php?name=poetics.butcher.index

[2] K. Ahrens *et al*. “Conceptual Metaphors: Ontology-based Representation and Corpora Driven Mapping Principles,” in the *Proc. of the ACL Workshop on the Lexicon and Figurative Language*. Sapporo, Japan, pp. 35–41, 2003.

[3] K. Ahrens *et al*. “From Lexical Semantics to Conceptual Metaphors: Mapping Principle Verification with WordNet and SUMO,” in Ji Donghong, Lua Kim Teng, and Wang Hui (Eds.), *Recent Advancement in Chinese Lexical Semantics: Proceedings of 5th Chinese Lexical Semantics Workshop (CLSW-5)*. Singapore: COLIPS, pp. 99–106, 2004.

[4] K. Ahrens *et al*. “Using WordNet and SUMO to Determine Source Domains of Conceptual Metaphors,” in Ji Donghong, Lua Kim Teng, and Wang Hui (Eds.), *Recent Advancement in Chinese Lexical Semantics: Proceedings of 5th Chinese Lexical Semantics Workshop (CLSW-5)*. Singapore: COLIPS, pp.91–98, 2004.

[5] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MIT Press, 2004.

[6] J. A. Barnden *et al*. (n.d.). *ATT-Meta Project: Metaphor, Metonymy and Mental States* [Online]. Available: http://www.cs.bham.ac.uk/~jab/ATT-Meta/

[7] J. A. Barnden and M.G. Lee, (2003) *Mental Metaphors from the Master Metaphor List: Empirical Examples and the Application of the ATT-Meta System* [Online]. Available: http://www.cs.bham.ac.uk/~jab/ATT-Meta/Papers/mml.CSRP-01-03.pdf

[8] E. Baumer, “Computational Metaphor Identification to Foster Critical Thinking and Creativity,” Ph.D. dissertation, Dept. Information and Computer Sci., Univ. of California, Irvine, 2009.

[9] E. Baumer *et al*. “Fostering metaphorical creativity using computational metaphor identification,” in Proc. Of the 7th ACM Conference on Creativity and Cognition, 2009.

[10] E. Baumer, J. P. White, B. Tomlinson, “Comparing Semantic Role Labeling with Typed Dependency Parsing in Computational Metaphor Identification,” in *Workshop on Computational Approaches to Linguistic Creativity* (CALC-10) at HLT/NAACL (Los Angeles, CA), 2010.

[11] J. Birke and A. Sarkar, “A Clustering Approach for the Nearly Unsupervised Recognition of Non-literal Language,” in Proceedings of EACL-06, pp. 329–336, 2006.

[12] E. Brill, “A Corpus Based Approach to Language,” Ph.D. dissertation, Dept. Computer Science, Univ. Pennsylvania, Philadelphia, 1993.

[13] B. Bowdle and D. Gentner, “The career of metaphor,” in Psychological Review [Online]. Available: http://www.psych.northwestern.edu/psych/people/faculty/gentner/newpdfpapers/BowdleGentnerInPress.pdf

[14] K. Burke, *Language As Symbolic Action: Essays on Life, Literature, and Method*, University of California Press, 1968.

[15] X. Carreras *et al*. “TAG, Dynamic Programming and the Perceptron for Efficient, Feature-rich Parsing,” in Proceedings of *Conference on Computational Natural Language Learning* (CONLL), Manchester, UK, 2008.

[16] J. M. Carroll and R. L. Mack, “Metaphor, Computing Systems, and Active Learning,” in *International Journal of Man-machine Studies* (IJMMS), vol. 22, no.1, pp. 39–57, 1985.

[17] Z. Chen *et al*. “Ontology Oriented Computation of English Verbs Metaphorical Trait,”, in Advances in Computational Linguistics (CICLing-2009). Research in Computing Science, vol. 41, 2009, pp. 135–142, 2009.

[18] S. Chung *et al*. “Source Domains as Concept Domains in Metaphorical Expressions,” in *Computational Linguistics and Chinese Language Processing*, vol. 10, no. 2005, 4, pp. 553–570.

[19] S. Chung and K. Ahrens, “Source Domain Determination: WordNet-SUMO and Collocation,” in the Proc. of *Theme Session of Cognitive-linguistic Approaches: What Can We Gain by Computational Treatment of Data?* The 2nd Int. Conf. of the German Cognitive Linguistics Association. pp. 1–4, 2006.

[20] M. Collins and N. Duffy, “Convolution Kernels for Natural Language,” in *Proceedings of Neural Information Processing Systems (NIPS) 14*, Vancouver, British Columbia, Canada, 2001.

[21] M. Collins and N. Duffy, “New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron,” in *Proceedings of Association of Computing Linguistics (ACL),* Philadelphia, PA, 2002.

[22] M. Collins, “Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with the Perceptron Algorithm,” in *Proceedings of Empirical Methods in Natural Language Processing (EMNLP),* Philadelphia, PA*, 2002*.

[23] M. Collins. (2002, Jan. 4). *Parameter Estimation for Statistical Parsing Models: Theory and Practice of Distribution-Free Methods* [Online]. Available: http://people.csail.mit.edu/mcollins/papers/book.ps

[24] M. Collins and T. Koo (2005). *Discriminative Reranking for Natural Language Parsing* [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.5338&rep=rep1&type=pdf

[25] M. Collins, "Head-driven Statistical Models for Natural Language Parsing". Ph.D. dissertation, Dept. Computer and Information Science, Univ. of Pennsylvania, Philadelphia, 1999.

[26] M. Collins, "New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pp. 263–270. Philadelphia, PA, 2002.

[27] M. Collins, "Ranking Algorithms for Named-Entity Extraction: Boosting and the Voted Perceptron," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, pp. 489–496. Philadelphia, PA, 2002.

[28] F. Costa *et al*., (2003). *Towards Incremental Parsing of Natural Language Using Recursive Neural Networks* [Online]. Available: http://www.dsi.unifi.it/~costa/online/costa_towards_03.pdf

[29] H. Cunningham *et al*., "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications." in *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics* (ACL'02). Philadelphia, July 2002.

[30] W. B. Dolan, "Metaphor as an Emergent Property of Machine-Readable Dictionaries," Microsoft Research, Microsoft Corp., Redmond, WA, Tech. Rep., MSR-TR-95-11, pp. 1, March 1995.

[31] R. Duda, P. Hart, and D. Stork, "Unsupervised Learning and Clustering" in *Pattern Classification* (2nd ed.) New York, Wiley-Interscience, 2001.

[32] B. Decadt and W. Daelemans, "Verb Classification – Machine Learning Experiments in Classifying Verbs into Semantic Classes," in *Proceedings of the 4th International Conference on Language Resource and Evaluation*, 2004.

[33] L. Fainsilber and A. Ortony, "Metaphor production in the description of emotional states," in Metaphor and Symbolic Activity, vol. 2, pp. 239–250, 1987.

[34] D. Fass, and Y. Wilks, "Preference Semantics, Ill- Formedness and Metaphor," in American Journal of Computational Lingnistics, vol. 9, no.3–4, pp. 178–187, 1983.

[35] D. Fass, "Collative Semantics," in Proceedings of the 11th conference on Computational Linguistics, 1986.

[36] D. Fass. “Collative Semantics,” in Proceedings of the 11th Conference on Computational linguistics (COLING '86). Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 341–343, 1986.

[37] D. Fass, “Met*: A Method for Discriminating Metonymy and Metaphor,” in *Computational Linguistics*, vol. 17, no. 1, Cambridge, MA, MIT Press, pp. 49–90,1991.

[38] C. J. Fillmore, "Scenes-and-frames semantics," in Linguistic Structures Processing, A. Zampolli, Ed., Amsterdam, North-Holland, pp. 55–81, 1977.

[39] C. J. Fillmore, "Frame semantics," In The Linguistic Society of Korea, eds. Linguistics in the Morning Calm. Seoul: Hanshin, pp.111–37,1982.

[40] S.K. Foss, *Rhetorical Criticism: Exploration and Practice*. 2nd edition. Long Grove, IL: Waveland Press, 1995.

[41] J. Geary, “All Shook Up,” in *I Is an Other: The Secret Life of Metaphor and How It Shapes the Way We See the World,* Temple Hills, MD: Harper Press, pp. 5–6, 2011.

[42] D. Gentner and J. Grudin, “The Evolution of Mental Metaphors in Psychology: A 90-Year Retrospective,” in *American Psychologist*, vol. 40, no. 2, pp.181–192,1985.

[43] D. Gentner *et al*., “Metaphor Is Like Analogy,” in The analogical mind: Perspectives from Cognitive Science, D. Gentner, K.J. Holyoak, B.N. Kokinov, B.N. (Eds.), Cambridge MA, MIT Press. pp.199–253, 2001.

[44] M. Gedigian *et al*. “Catching Metaphors,” in *Proceedings of the Third Workshop on Scalable Natural Language Understanding* (ScaNaLU 06) Issue: June, Publisher: Association for Computational Linguistics, pp. 41–48, 2006.

[45] R. Gibbs, “Literal meaning and psychological theory,” in *Cognitive Science*, vol. 8, pp.275–304, 1984.

[46] R. Gibbs and M. Tendahl, “Cognitive Effort and Effects in Metaphor Comprehension: Relevance Theory and Psycholinguistics,“ in *Mind and Language*, vol. 21, no. 3, pp. 379–403, 2006.

[47] R.W. Gibbs, “When is Metaphor? The Idea of Understanding in Theories of Metaphor,” in Poetics Today, vol. 13, no. 4, Aspects of Metaphor Comprehension, pp. 575–606, 1992.

[48] M. Hart (1974), *Project Gutenberg* [Online]. Available: http://www.gutenberg.org/wiki/Main_Page.

[49] X. Huang et al, “Computational Methods for Metaphor in Language; A Survey,” in *Journal of Computer Science and Technology*, vol. 22, no.2, pp. 308–319, 2006.

[50] S.I. Hayakawa, Preface in “Language in Thought and Action.” 5th ed. (New York: Harcourt, Brace, Jovanovich, 1978): pp xiii.

[51] A. Kempe, (1997). *Finite State Transducers Approximating Hidden Markov Models* [Online]. Available: http://acl.ldc.upenn.edu/P/P97/P97-1059.pdf

[52] D. Klein and C. Manning. “Factored A* Search for Models over Sequences and Trees,” in Proceedings of the *International Joint Conference on Artificial Intelligence* (IJCAI), Acapulco, Mexico 2003.

[53] D. Klein and C. Manning, “A* Parsing: Fast Exact Viterbi Parse Selection,” in Proceedings of the *North American Chapter of the Association for Computational Linguistics* (NAACL), Edmonton, Alberta, Canada 2003.

[54] D. Klein, “The Unsupervised Learning of Natural Language Structure,” Ph.D. dissertation, Dept. Computer Science, Stanford University, Stanford, CA, 2005.

[55] T. Koo *et al*. “Simple Semi-supervised Dependency Parsing,” in Proceedings of *Association for Computational Linguistics* (ACL), OH, 2008.

[56] Z. Kovecses, Preface to the Second Edition in “Metaphor a Practical Introduction,” 2nd ed. (Oxford University Press Inc., New York, NY, 2010.

[57] J.L. Krieger *et al*. “Metaphor use and health literacy: A pilot study of strategies to explain randomization in cancer clinical trials,” in *Journal of Health Communication: International Perspectives*, vol. 1, no. 16, pp. 3–16, 2011.

[58] S. Krishmakumaran and X. Zhu, (2007). “Hunting Elusive Metaphors Using Lexical Resources,” in *Computational Approaches to Figurative Language*, Workshop at HLT/NAACL 2007. Rochester, NY.

[59] S. Kullback and R. A. Leibler, “On Information and Sufficiency,” in Annals of Mathematical Statistics, col. 22, no. 1, pp. 79–86, 1951.

[60] G. Lakoff and M. Johnson, *Metaphors We Live By*. Chicago, University of Chicago Press, 1980.

[61] G. Lakoff and M. Johnson, *Metaphors We Live By*. Chicago, University of Chicago Press, pp. 155–156, 1980.

[62] G. Lakoff *et al*. 1991. The master metaphor list. Technical report, University of California at Berkeley.

[63] J. Lawley and P. Tompkins, “Metaphors in Mind: Transformation through Symbolic Modeling,” Developing Company Press, October, 2000.

[64] G. Luger, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving (6th ed.)* Boston, MA, Addison-Wesley, 2009.

[65] G. Mann, “An Unsupervised Learning Method for Metaphor Detection”, Honors Thesis, Dept. Comput. Sci., Brown University, Providence, RI, 1999.

[66] C. Manning and H. Schütze, “Evaluation measures,” in *Foundations of statistical natural language processing*, chapter 8.1, Cambridge, MA, MIT Press, pp 267– 271, 1999.

[67] C. Manning and H. Schütze, “Selectional Preferences,” in *Foundations of statistical natural language processing*, chapter 8.4, Cambridge, MA, MIT Press, pp 288– 294, 1999.

[68] L. Marquez *et al*. (1998). *A Machine Learning Approach to POS Tagging* [Online]. Available: http://www.lsi.upc.es/~nlp/papers/report98-mpr.ps.gz

[69] M. Marcus *et al*. “Building a Large Annotated Corpus of English: The Penn Treebank,” in *Computational Linguistics, 19*, pp. 313–330, 1993.

[70] J. H. Martin, *A Computational Model of Metaphor Interpretation*, Academic Press Professional, Inc., San Diego, CA, USA.1990.

[71] J. H. Martin, “Metabank: A knowledge base of metaphoric language conventions,” in *Computational Intelligence*, vol. 10, no. 2, pp. 134–149, 1994.

[72] J.H. Martin, “Computational Approaches to Figurative Language,” in Metaphor and Symbolic Activity, vol. 11, no.1, pp. 85–100, 1996.

[73] Z. Mason, “CorMet: a computational, corpus-based conventional metaphor extraction system,” Computational Linguistic, Volume 30 Issue 1, March 2004 MIT Press Cambridge, MA, USA, pp. 23–44, 2004.

[74] Z. Mason, “CorMet: a computational, corpus-based conventional metaphor extraction system,” Computational Linguistic, Volume 30 Issue 1, March 2004 MIT Press Cambridge, MA, USA, pp. 43, 2004.

[75] G. Miller, (2006) *WordNet: An Electronic Lexical Database*, Bradford Books, copyright Princeton University, [Online]. Available: http://wordnet.princeton.edu/perl/webwn?s=word-you-want

[76] Min, X. (2005) "Metaphors." Beyond Intractability. Eds. Guy Burgess and Heidi Burgess. Conflict Research Consortium, University of Colorado, Boulder. [Online]. Available: http://www.beyondintractability.org/essay/metaphors/

[77] R. Mooney, "Learning for Semantic Parsing," in proceedings of *Computational Linguistics and Intelligent Text Processing: Proceedings of the 8th International Conference, CICLing 2007, Mexico City* (invited paper), A. Gelbukh (Ed.), pp. 311–324, Springer, Berlin, Germany, February 2007.

[78] I. Niles *et al*. "Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology," in *Proceedings of the 2003 International Conference on Information and Knowledge Engineering* (IKE 03), Las Vegas, NV, 2003.

[79] A. Ortony, "Why Metaphors Are Necessary and Not Just Nice," in Educational Theory, vol. 25, no. 1, pp. 45–53, 1975.

[80] A. Ortony, "Some Psycholinguistic Aspects of Metaphor," in Cognition and Figurative Language, Hillsdale, NJ: Erlbaum Associates, pp. 69–83, 1980.

[81] B. Pasanek and D. Sculley, "Mining Millions of Metaphors," in Oxford Journal *Literary and Linguistic Computing*, vol. 23 no. 3, pp. 345–360, 2008.

[82] B. Pasanek. (2006). *The Mind is a Metaphor* [Online]. Available: http://mind.textdriven.com/db/browse.php

[83] A. Pauls and D. Klein, "K-Best A* Parsing", in the Proc. of *Association for Computational Linguistics*, Singapore, 2009.

[84] A. Pauls *et al*. "Top-Down K-Best A* Parsing", in the Proc. of *Association for Computational Linguistics*, Sweden, 2010.

[85] S. Petrov and D. Klein, "Learning and Inference for Hierarchical Split PCFGs", in proceedings of *Advancement for the Advancements of Artificial Intelligence* AAAI, Vancouver, British Columbia, 2007.

[86] S. Petrov, "Coarse to Fine Natural Language Processing," Ph.D. dissertation, Dept. Comput. Sci., Univ. California, Berkeley, 2009.

[87] R. Quirk, S. Green. *A Students Grammar of the English Language.* New York, Longman Publishing Group, 1990.

[88] P. Resnik, “Selection and Information: A Class- based Approach to Lexical Relationships,” Ph.D. dissertation, Dept. Comput. and Inform. Sci., Univ. of Pennsylvania, Philadelphia, 1993.

[89] J. Ruppenhofer et al (2010), *FrameNet II: Extended Theory and Practice* [Online]. Available: http://www.icsi.berkeley.edu/framenet

[90] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (2nd ed.)* Upper Saddle River, Prentice Hall, 2009.

[91] I. A. Richards, “Lecture V: Metaphor,” in *The Philosophy of Rhetoric*, New York, Oxford University Press, pp. 89–136, 1936.

[92] B. Scholkopf and A. Smola, *Learning with Kernels*. Cambridge, MIT Press, 2002.

[93] K. Schuler, “VerbNet: A broad-coverage, comprehensive verb lexicon,” Ph.D. dissertation, Dept. Computer and Information Science, Univ. Pennsylvania, Philadelphia, 2005.

[94] S. Skiena, *The Algorithm Design Manual*. New York, Springer, 2009.

[95] L. Shi, and R. Mihalcea, “Putting the pieces together: Combining FrameNet, VerbNet, and WordNet for robust semantic parsing,” in Proceedings of the 6th *International Conference on Intelligent Text Processing and Computational Linguistics* (CICLing), Mexico City, Mexico. 2005.

[96] L. Shi and R. Mihalcea, “An Algorithm for Open Text Semantic Parsing,” in Proceedings of the 3rd workshop on *Robust Methods in Analysis of Natural language Data* (ROMAND), Geneva, 2004.

[97] E. Shutova, “Models of Metaphor in NLP,” in Proceedings of the 48th *Annual Meeting of the Association of Computational Linguistics* (ACL), Uppsala, Sweden, pp 688–697, July, 2010.

[98] E. Sommer and D. Weiss, *Metaphor’s Dictionary*, First Edition. "Metaphor." International Thomson Publishing Company. pp. vii–x, 1995.

[99] G. Steen *et al*. “MIP: A Method for Identifying Metaphorically Used Words in Discourse,” in Metaphor and Symbol, vol. 22, no.1, pp.1–39, 2007.

[100] X. Tang et al, "Automatic Metaphor Recognition Based on Semantic Relation Patterns," in Asian Language Processing (IALP), 2010 International Conference on Asian Language Processing, pp.95-100, Dec. 2010

[101] X. Tang *et al*. "Beyond Selectional Preference: Metaphor Recognition with Semantic Relation Patterns,” in International Journal of Asian Language Processing, Vol. 20, no. 4, pp.141–156, Dec. 2010.

[102] T. Wand *et al*. “Extracting a Domain Ontology from Linguistic Resource Based on Relatedness Measurements,” in *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence,* Compiegne, France, pp. 345–351, 2005.

[103] E. Weiner, “A Knowledge Representation Approach to Understanding Metaphors,” in *Computational Linguistics*, vol. 10, no. 1, Cambridge, MA, MIT Press, pp. 1–14, 1984.

[104] Y. Wilks, “A Preferential Pattern-seeking Semantics for Natural Language Inference,” in *Artificial Intelligence*, vol. 6, no. 1, 1975, pp 53–74.

[105] Y. Wilks, (1975) “An intelligent analyzer and understander of English”, in Comm. Of the ACM, vol. 18, no. 5, 1975.

[106] Y. Wilks, (1977) "Good and Bad Arguments about Semantic Primitives," in Springer Words and Intelligence, 6, pp. 103–139, 2007.

[107] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques (2nd ed.)* Morgan Kaufman, 2005.

[108] L. Yaoyong. (2006). *Hierarchical, Perceptron-like Learning for Ontology-Based Information Extraction* in *16th International World Wide Web Conference (WWW2007)* [Online]. Available: http://gate.ac.uk/sale/www07/www428-li.pdf

[109] L. Yaoyong et al, “Using Uneven Margins SVM and Perceptron for Information Extraction,” in *Proceedings of Ninth Conference on Computational Natural Language Learning (CoNLL-2005),* Ann Arbor, MI, 2005.

[110] X. Zhu. (2008, July 19). *Semi-supervised learning literature survey. Technical Report* [Online]. Available: http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf

[111] C. Zhou et al, “Computational Mechanisms for Metaphor in Language: A Survey,” in Journal of Computer Science and Technology, Col. 22, no. 2, pp 308–319, 2007.

BIOGRAPHICAL SKETCH

Daniel Nieten received his Bachelor of Science degree in computer and information science and engineering at the University of Florida in 1992, and his Master of Science in computer engineering at the University of Florida in 2004. He has worked on transactional systems; on command and control systems in energy management, ground control system for human spaceflight; and telecommunications. He is currently a Director with Modus Operandi, Inc., in Melbourne, FL working with natural language processing. Daniel is married to Teresa Nieten, PhD and has a son, Brandon.