Story Detection Using Generalized Concepts

by

Nitesh Kedia

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2015 by the
Graduate Supervisory Committee:

Hasan Davulcu, Chair
Steven Corman
Baoxin Li

ARIZONA STATE UNIVERSITY

May 2015

ABSTRACT

A major challenge in automated text analysis is that different words are used for related concepts. Analyzing text at the surface level would treat related concepts (i.e. actors, actions, targets, and victims) as different objects, potentially missing common narrative patterns. Generalized concepts are used to overcome this problem. Generalization may result into word sense disambiguation failing to find similarity. This is addressed by taking into account contextual synonyms. Concept discovery based on contextual synonyms reveal information about the semantic roles of the words leading to concepts. Merger engine generalize the concepts so that it can be used as features in learning algorithms.

DEDICATION

I dedicate my dissertation work to my family and friends. A special thanks to my friend Tejas M.U, whose has always been an encouragement. She has always been ready to question and answer the problems I faced. I also dedicate this dissertation to my friends Satyabrata Sharma and Nishant Bansal who have supported me throughout the process. I will always appreciate all they have done.

ACKNOWLEDGMENTS

I sincerely thank my advisor Dr. Hasan Davulcu for his continued guidance, support and encouragement during my masters and while writing this thesis. My sincere thanks to Prof. Steven Corman, to provide me an opportunity to work under his guidance, who has encouraged and guided me throughout the process. I also would like to thank Dr. Baoxin Li for being on my thesis supervisory committee. I would like to thanks all my lab-mates Betul, Nyunsu, Anvesh, Pranay, Shravan, Sultan and all members of Centre for Strategic Communication (CSC) research lab of Hugh Downs School of Human Communication at Arizona State University for making this journey so exciting and making this a great learning experience. I am very grateful for the love and the unconditional support of my family and friends.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF ALGORITHMS

Algorithm Page

CHAPTER 1

INTRODUCTION

1.1 Statement of Purpose

Text classification is a type of machine learning approach where data is classified into pre-defined classes e.g. classifying an email as spam or non-spam. We use algorithmic text classification to reduce the human effort in classifying the data. There are two main types of machine learning approach.

- Supervised Learning
- Unsupervised Learning

Supervised learning involves two phases namely training and testing phases. In training phase a model is trained with correctly labeled data. This model is then used to classify the testing data and measure the accuracy of the model. Supervised learning is mainly used in the scenarios where we know the data contains the class labels which can be used to build the training sample using the knowledge of domain experts whereas when the number or types of class labels are not very clear then unsupervised learning comes at rescue. In unsupervised learning we cluster the data into separate groups by selecting features and techniques which can effectively lead to meaningful clusters.

In this thesis, we use supervised learning technique to predict class of unlabeled data based on pre classified training sample. Since good feature extraction is a very important step towards an accurate machine learning approach, our research focus is on discovering features which we call generalized concepts that are generated by clustering triplets extracted from the paragraphs.

There are 3 major steps involved in this.

1. Syntactic merging criteria.
2. Contextual Similarity as a threshold while merging.
3. Bottom-up merging algorithm.

We define a triplet in a sentence as a relation between subject and object, the relation being the predicate (verb). Extraction of Triplets is a process of finding significant information from an input text like subject (who), verb (doing what), direct object (to whom), Indirect Object (when and where). Triplet extraction, in addition removes irrelevant information such as stop words (a, an, the, he, she, etc) and irrelevant clauses.

Triplets are generated separately for stories and Non-Stories. After the triplets are generated, it is cleaned using normal TF-TFD measure using which triplets containing verbs like 'be', 'say', 'kill' are removed from the training set. These triplets are then passed through the above mentioned steps to generate the generalized concepts for the two classes. Please refer chapter 4 for more details on this.

1.2 Scope

Scope of this thesis research is to classify text documents into two major classes i.e. Story and Non-Story. In chapter 2, we elaborate the significance, definition and background of these two categories. We use a corpus of 39642 paragraphs where 9058 Paragraphs are coded as stories and 37584 coded as Non-stories by the domain experts to develop this story classifier. Training data is a collection of Islamist extremist texts, speeches, video transcripts, forum posts, etc., collected in open source.

1.3 Motivation

An investigation of terrorist narrative communication through an in-depth examination of extremists published autobiographies and interviews can be helpful in understanding mindsets and motivation behind terrorist activities. To detect the relevant content from the large amount of data available we need to build a story classifier.

Our study is motivated by the observation [1] that interrelated stories that work together as a system are fundamental building blocks of (meta-) narrative analysis. We focus on discriminating between stories, and non-stories. The main purpose of developing an automated story classifier is to reduce the human dependency to annotate story and non-stories.

The main contribution of this thesis is the introduction of a new set of generalized concepts which are generated by clustering <subject, verb, and object> triplets.

1.4 Outline

The rest of the report is organized as follows.

Chapter 2 gives Background Literature.

Chapter 3 mentions Related Work.

Chapter 4 describes System Architecture.

CHAPTER 2

BACKGROUND LITERATURE

Personal narratives are powerful sources of persuasion, none more so than stories than those cultural heroes tell about their own lives [2]. Whether their account retells the story of a great athlete or actor or celebrity or terrorist, fans are drawn to these accounts as moths to bright lights. In part this is because the stories themselves can be quite interesting, and in part because readers often closely want to in some way identify their own lives with the life stories of their heroes [3]. An investigation of terrorist narrative communication through an in-depth examination of extremists published autobiographies and interviews can be helpful in understanding mindsets and motivation behind terrorist activities. In addition, the analysis of terrorist narratives across geographical regions holds the potential to illustrate cultural differences, as well as to illustrate how telling their own stories serve to recruit and assimilate outsiders into local political groups and extremist organizations. But the problem with analysis of extremist text is that it needs many human annotators to extract stories and non-stories from different sources. The main purpose behind story classifier module is to remove the human dependency to annotate story and non-stories.

A story is comprised of three components. First, there must be an actor or actors telling the story implicitly or explicitly. This can include politicians, mujahedeen, and everyday people and so on. Second, the actors must be performing actions. This can include fighting, preparing for a battle, talking to others and soon. Third, the actor's actions must result in a resolution. Resolutions can include a new state of affairs, a new equilibrium created, a previous equilibrium restored, and victory and so on. Besides, stories usually have story worlds, or worlds were the stories are taking place.

Story worlds are not fictional universes, but rather environments in which the story takes place.

Story Example: ”They have planted your remains in the sands like a flag To motivate the people morning and night Oh, woe unto them, they have raised a beacon of blood To inspire tomorrow’s generation with hate and dislike”. A non-story paragraph is one, among the categories Exposition, Supplication, Question, Annotation, Imperative and Verse.

Non-Story Example: “Let the soldiers of this Administration go to hell. Petraeus and Bush are trying to convince the Americans that their salvation will begin six weeks from next July. In fact even if Bush keeps all his forces in Iraq until doomsday and until they go to hell, they will face only defeat and incur loss, God willing.” This paragraph is coded as “Non-Story” because there is no explicit resolution. There are only hypothetical resolutions.

CHAPTER 3

RELATED WORK

Improved unsupervised name discrimination with very wide bigrams and automatic cluster stopping, Ted Pedersen. They develop an unsupervised approach to name discrimination where numbers of clusters are automatically determined.

Computational models of stories have been studied for many different purposes. R.E. Hoffman et al. (2011) [4] modeled stories using an artificial neural network. After the learning stage, they compare the story- recall performance of the neural network with that of schizophrenic patients as well as normal controls in order to derive a computational model which matches the illness mechanism. The most common form of classification applied for stories tackles the problem of mapping a set of stories to predefined categories. One of the popular applications is the classification of news stories to their topics [5], [6]. Gordon investigated the problem of detecting stories in conversational speech [7] and weblogs [8] and [9]. In [7], the authors train a Naive Bayes classifier to categorize the transcribed text of a speech into story and non-story categories. Using word-level unigram and bigram frequency counts as feature vectors, they reported results for the classification of a speech as a story with 53.0% precision, 62.9% recall and 0.575 F-measure. For weblogs, in [8], they incorporated techniques for automatically detecting sentence boundaries to their previously used text features to train a Support Vector Machine classifier. After smoothing the confidence values with a Gaussian function, they achieved 46.4% precision, 60.6% recall and 0.509 F-measure. In Gordon and Swanson's most recent work on story classification [9], they used a confidence-weighted linear classifier with a variety of lexical features, and obtained the best performance with unigrams. They applied this classifier to classify weblog posts in 9 the ICWSM 2009 Spinn3r Dataset, and they obtained 66% precision, 48% recall, and F-measure of 0.55.

CHAPTER 4

SYSTEM ARCHITECTURE

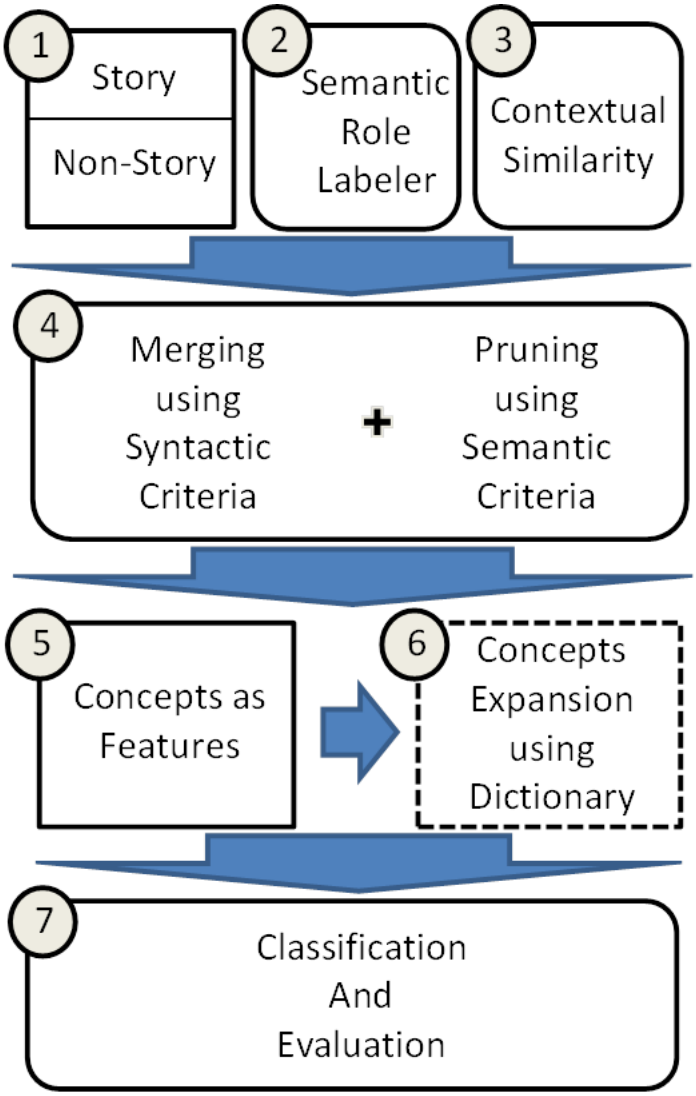

Figure 1: System Architecture

4.1 Data Collection

We use a corpus of 39642 paragraphs where 9058 Paragraphs are coded as Stories and 30584 coded as Non-Stories by the domain experts to develop this story classifier. Text is collected from the Islamic extremist from sources such as al-Qaeda, ISIS and related groups which sympathize with its cause and methods. Area specialists selected documents which they consider would contain stories, defined as order of associated events, leading to a purpose or projected purpose.

Extremists' texts are not fully composed of stories, and that is why the purpose of this project is to detect the portion of texts that are stories. To help this domain expert developed eight mutually exclusive and exhaustive categories namely stories, question, verse, supplication, imperative, exposition, annotation and others with definitions and examples on which coders could be trained and made to label the collected data.

4.2 Semantic Role Labeler

We follow a standard verb-based approach to extract the simple clauses within a sentence. A sentence is identified to be complex if it contains more than one verb. A simple sentence is identified to be one with a subject, a verb, with objects and their modifying phrases. A complex sentence involves many verbs. We define a triplet in a sentence as a relationship between a verb, its subject and object(s). Extraction of triplets [10][11][12] is the process of finding who (subject), is doing what (verb) with/to whom (direct objects), when and where (indirect objects/and prepositions).

4.3 Contextual Synonyms

Algorithm 1:

Given: Set of Triplets {s-v-o} ≡ T, s ϵ {S}, v ϵ {V}, o ϵ {O}

1. Create empty set {C}
2. Compute unique sets of {s-v}, {v-o} and {s-o}

   ∃ s ϵ {S}, v ϵ {V}, o ϵ {O} and s-v-o ϵ T
3. For every pair of s-v. Create and merge in{C}, s-v-{O},

   ∃ o ϵ {O}, s-v-o ϵ T
4. Similarly, for every pair of v-o, Create and merge in {C}, {S}-v-o,

   ∃ s ϵ {S}, s-v-o ϵ T.
5. Similarly, for every pair of s-o, Create and merge in {C}, s-{V}-o,

   ∃ v ϵ {V}, s-v-o ϵ T.
6. Create Similarity matrices:

   SubSim ≡ |S|*|S|, VerbSim ≡ |V|*|V| and ObjSim ≡ |O|*|O|
7. Loop through {C} ≡ {S}-{V}-{O}
   i. If Concept c ≡ {S}-v-o, For every pair of s1,s2 ϵ {S},

      SubSim [s1, s2] += 1

      Else if Concept c ≡ s-{V}-o, for every pair of v1, v2 ϵ {V},

      VerbSim [v1, v2] += 1

      Else if Concept c ≡ s-v-{O}, for every pair of o1, o2 ϵ {O},

      ObjSim [o1, o2] += 1

mujahideen	attack	base
mujahidin	storm	area
group	hit	house
soldier	seize	area
force	loot	home
lion	raid	station
hero	shoot	center
fighter	ambush	checkpoint
mujahid	assassinate	headquarters
brigade	bomb	land
mujahedeen	capture	location
detachment	disrupt	region

Table 1. Top Ten Similar Words For Mujahideen, Attack, Base

4.4 Merging Engine: Concept Generation

Concept generation is the most important part of this thesis research. To start with each triplet belongs to the concept set which is then merged in a bottom-up fashion to get richer concepts. Here we face the major challenges like complexity of the algorithm and maintaining the relevance of the concepts while merging. Firstly, the set of triplets are first cleaned as mentioned above and then the novel similarity algorithm based on bigrams from <Subject, Verb, Object> triplets (described in 4.3) were used to find the similarity between subject-subject, verb-verb and object-object pairs. This similarity is used to put a threshold while merging the concepts to eliminate outliers. A syntactic criterion described below is applied while merging the concepts which is finding common context between two concepts and merging them.

Algorithm 2: Syntactic Criteria

Given, C1 = {S1}-{V1}-{O1} and C2 = {S2}-{V2}-{O2}
We merge C1 and C2 if we meet the below criteria,
1. if {{S1} ≠{S2} and {V1} ∩ {V2} ≠ {} and {O1}∩{O2} ≠ {}} OR {S1} ≡{S2} and
2. if {{V1} ≠{V2} and {S1} ∩ {S2} ≠ {} and {O1}∩{O2} ≠ {}} OR {V1} ≡{V2} and
3. if {{O1} ≠{O2} and {S1} ∩ {S2} ≠ {} and {V1}∩{V2} ≠ {}} OR {O1} ≡{O2}

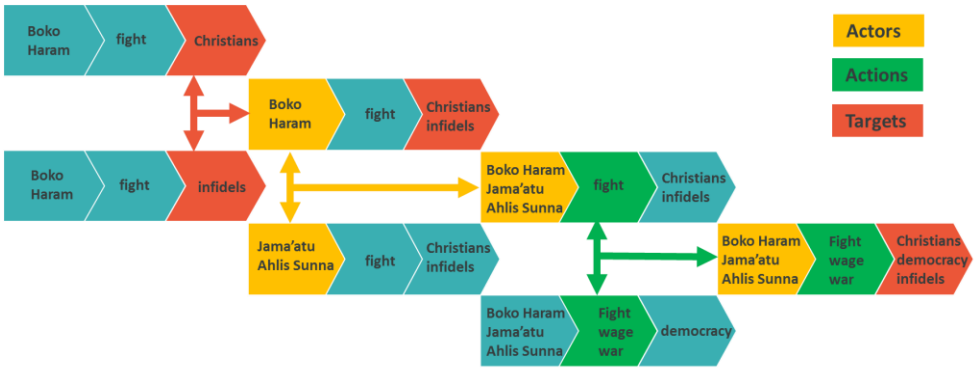

Fig 2: Syntactic Criteria

Algorithm 3: Merger Engine

1. Input:
    a. Set of Triplets { s-v-o } ≡ T
    b. Semantic similarity between all pairs of sϵ{S}, vϵ {V}, oϵ {O}.

2. Bottom Up Merging:
    a. Loop while flag == 1
        i. flag =0
        ii. Loop through cϵ{C} from contextual synonyms algorithm
            1. For Concept c find the matching concepts {Y} using *Syntactic Criteria*
            2. if size of {Y} >0,
                a. flag =1
                b. For each Concept yϵ {Y}
                    i. Merge y into c
                    ii. Prune c using *Semantic Criteria*
                    iii. Remove y from {C}

4.5 Classification and Evaluation

4.5.1 Concepts as Features

We use the set of 39642 paragraphs to train the classifier. Although we generated concepts separately for stories and non-stories, while building the feature matrix all the concepts are aggregated and the feature matrix is filled by matching paragraphs with concepts. For a paragraph to match a concept we take all the triplets present in that paragraph and look at the concepts containing those triplets. We train different classifiers using a set of 5835 Story concepts and 17702 Non-Story concepts. We get the best accuracy with SLEP-LogisticR classifier [10]. Refer below table for the accuracy of different classifiers.

Classifier	Precision	Recall	F-Measure
SVM	0.902	0.702	0.789
SLEP-LeastR	0.889	0.715	0.793
SLEP-LogisitcR	**0.870**	**0.838**	**0.854**

Table 2. Tier1: Story Accuracy

Classifier	Precision	Recall	F-Measure
SVM	0.757	0.922	0.831
SLEP-LeastR	0.764	0.905	0.884
SLEP-LogisitcR	**0.842**	**0.884**	**0.862**

Table 3. Tier1: Non-Story Accuracy

4.5.2 Logistic Regression

Logistic regression is basically a probabilistic classification model used to classify data into binary classes. Since we are dealing with two majority classes here this is best suited for our purpose. We use SLEP package [13] developed at Arizona State University by Dr. Jieping Ye and team to model this. This package provides a regularization parameter which can be passed to the function using parameter "rho".

4.5.3 Feature Extraction

As we can see from the table 1 and Table 2 that we get a good accuracy with the above set of concepts but the model is over fitted with 23000 features at this stage and we try to overcome that by using logistic regression with regularization. We can observe from graphs in Fig1 and Fig 2 that there is a sharp drop between the number of features between 12000 to 2000 but the accuracy is preserved. Using this we find the optimal number of features to be 6186 which solves the problem of over fitting.

Number of Features	Precision	Recall	F-Measure
23000	0.870	0.838	0.854
12576	0.868	0.847	0.857
6186	**0.874**	**0.783**	**0.826**
5659	0.875	0.746	0.805

Table 4. Story, Feature Extraction - Logistic Regression

Number of Features	Precision	Recall	F-Measure
23000	0.842	0.884	0.862
12576	0.851	0.871	0.861
6186	**0.803**	**0.887**	**0.843**
5659	0.779	0.893	0.832

Table 5. Non-Story, Feature Extraction - Logistic Regression

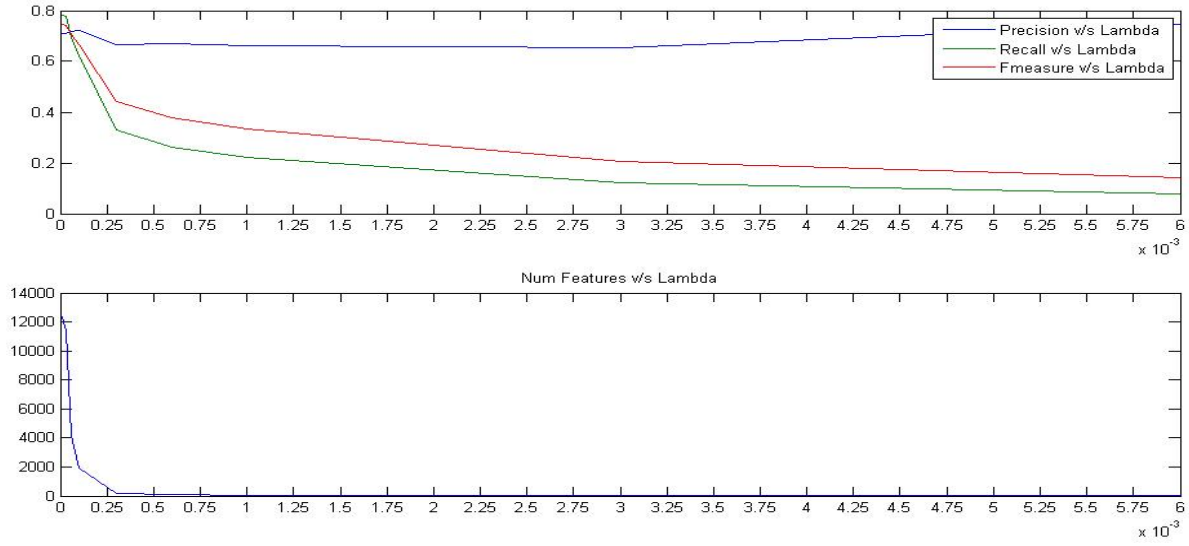

Fig 3. Story, Features V/S Accuracy - Logistic Regression

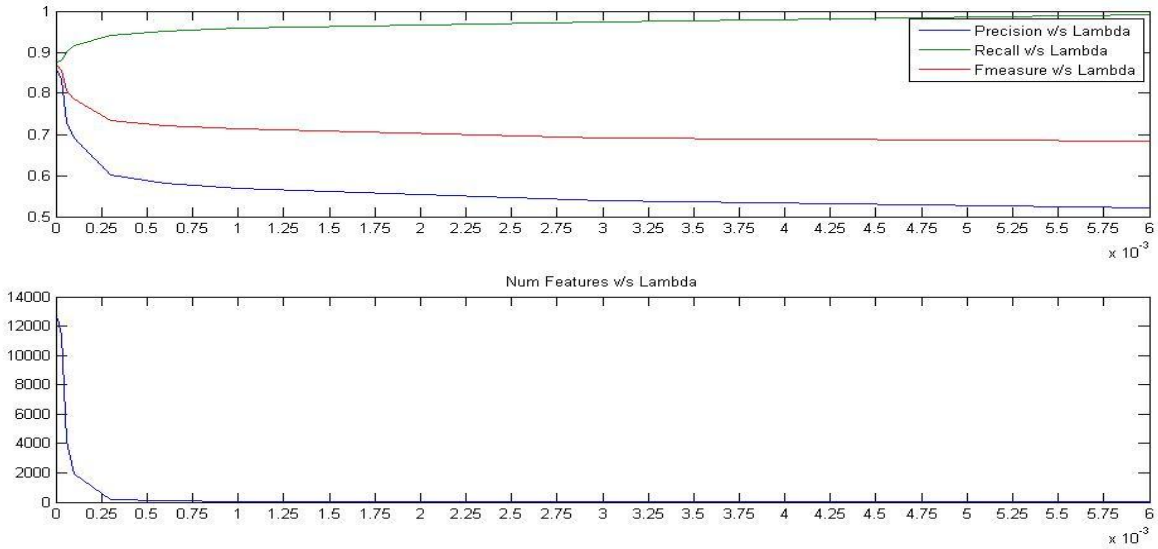

Fig 4. Non-Story, Features V/S Accuracy - Logistic Regression

4.5.4 Concept Expansion using Dictionary

We call the concepts generated from the Merger Engine as Tier 1 concepts. After that we do lateral expansion of concepts resulting from this bottom up algorithm using 3 techniques:

1. Expanding using similarity measure.
2. Expanding using WordNet.
3. Expanding using both similarity measure and WordNet.

We could observe a boost of 1% in F-measure with all the above techniques. Below accuracy is after we do feature extraction (Refer 6.4.1)

Method	Precision	Recall	F-Measure
Tier 1	0.874	0.783	0.826
Tier 1 + Similarity	0.856	0.818	0.836
Tier 1 + WordNet	0.873	0.800	0.835
Tier 1 + Similarity + WordNet	0.858	0.817	0.837

Table 6. Story Accuracy After Lateral Expansion of Concepts

Method	Precision	Recall	F-Measure
Tier 1	0.803	0.887	0.843
Tier 1 + Similarity	0.825	0.862	0.843
Tier 1 + WordNet	0.815	0.883	0.848
Tier 1 + Similarity + WordNet	0.825	0.865	0.845

Table 7. Non-Story Accuracy After Lateral Expansion of Concepts

4.5.5 K-Fold Cross Validation

We are doing a supervised learning here and to measure the accuracy of the model developed above we apply k-fold cross validation technique where k =10. We divide out training sample into 10 buckets and build the model from the data aggregated from 9 buckets. We used this model to test the 10^{th} bucket and save the accuracy. We repeat this by using each bucket once as the testing bucket. Multiple iterations are used to different partitions to reduce the variability, and the results are averaged over the iterations. All the above results are calculated using this technique.

CHAPTER 4

FUTURE WORK

Next major challenge to be faced is to find the pattern between discovered concepts and how they are related. We will be working on clustering these concepts to get meaningful results.

Also, there is a need to visualize these clusters effectively so that they can be analyzed by area experts.

REFERENCES

[1] H. L. Halverson, J. R. Goodall and S. R. Corman, Master Narratives of Islamist Extremism. New York: Palgrave Macmillan, 2011.

[2] J. Bruner and S. Weisser, "Autobiography and the construction of self," 1992.

[3] C. Joseph, The hero with a thousand faces. Princeton University Press, 1949.

[4] R. Hoffman, U. Grasemann, R. Gueorguieva, D. Quinlan, D. Lane, and R. Miikkulainen, "Using computational patients to evaluate illness mechanisms in schizophrenia," Biological psychiatry, 2011.

[5] B. Masand, G. Linoff, and D. Waltz, "Classifying news stories using memory based reasoning," in Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1992, pp. 59–65.

[6] D. Billsus and M. Pazzani, "A hybrid user model for news story classification," Lectures-International Centre for Mechanical Sciences, pp. 99–108, 1999.

[7] A. S. Gordon and K. Ganesan, "Automated story capture from conversational speech," in K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture, ACM. Banff, Canada: ACM, 2005, p. 145–152.

[8] A. Gordon, Q. Cao, and R. Swanson, "Automated story capture from internet weblogs," in Proceedings of the 4th international conference on Knowledge capture. ACM, 2007, pp. 167–168.

[9] A. Gordon and R. Swanson, "Identifying personal stories in millions of weblog entries," in Third International Conference on Weblogs and Social Media, Data Challenge Workshop, San Jose, CA, 2009.

[10] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic, "Triplet extraction from sentences," ´ Proceedings of the 10th International Multiconference Information Society-IS, pp. 8–12, 2007.

[11] D. Hooge Jr, "Extraction and indexing of triplet-based knowledge using natural language processing," Ph.D. dissertation, University of Georgia, 2007.

[12] Siddhartha Jonnalagadda, "An Effective Approach to Biomedical Information Extraction with Limited", Ph.D. dissertation, Arizona State University, 2011.

[13] J. Liu, S. Ji, and J. Ye. SLEP: SparseLearning with Efficient Projections. Arizona State University, 2009. http://www.public.asu.edu/~jye02/ Software/SLEP.