2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Story Detection Using Generalized Concepts and Relations

*Betul Ceran**, *Nitesh Kedia**, *Steven R. Corman*† *and Hasan Davulcu**
**School of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287-8809*
Email: {betul, nitesh.kedia, hdavulcu}@asu.edu
†*Hugh Downs School of Human Communication, Arizona State University, Tempe, AZ 85287-1205*
Email: steve.corman@asu.edu

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. Shallow parsers reveal semantic roles of words leading to subject-verb-object triplets. We developed a novel algorithm to extract information from triplets by clustering them into generalized concepts by utilizing syntactic criteria based on common contexts and semantic corpus-based statistical criteria based on "contextual synonyms". We show that generalized concepts representation of text (1) overcomes surface level differences (which arise when different keywords are used for related concepts) without drift, (2) leads to a higher-level semantic network representation of related stories, and (3) when used as features, they yield a significant 36% boost in performance for the story detection task.**

1. Introduction

Extremist groups use stories to frame contemporary events and persuade audiences to adopt their extremist ideology. Foreign policy of most countries in the twenty first century is centrally influenced by perceptions, attitudes and beliefs of societies. Therefore, it's of critical importance to fully understand the means by which extremist groups leverage cultural narratives in support of their ideological agenda. Understanding the structure of extremist discourse will provide better intelligence on what kinds of narrative and persuasive appeals they are making, allowing both better detection of trends and better knowledge of which themes might best be countered and how this might be accomplished. The research presented in this paper mainly focuses on extracting high-level relations and concepts which are then utilized for detecting stories and themes embedded in longer messages of extremist discourse.

A major challenge facing automated discourse analysis is that word usage can differ between two texts even though they are talking about the same thing. For example, violent extremists may use words such as *"brothers"*, *"mujahidin"*, *"mujahedeen"* and even *"lions of Islam"* to refer to the same group of people. 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. We address this problem by discovering *"contextual synonyms"* [1] which are verb and noun phrases that occur in similar contexts. After revealing contextual similarity, we generalize such references to a common node in a semantic network.

We developed an unsupervised and domain-independent framework which extracts high-level information from text as relationships and concepts forming a semantic network. It first uses a semantic role labeler to obtain ground facts as semantic triplets from text, and then proceeds to generalize them through a bottom-up agglomerative clustering algorithm. Semantic role labeling, i.e. shallow semantic parsing, is a task in natural language processing which recognizes the predicate or *verb phrases* in a sentence along with its semantic arguments and classifies them into their specific roles as *subjects* and *objects*. For example, we would like to merge extracted triplets such as ⟨*mujahidin*→*kill*→*kafir*⟩ and ⟨*ISIS* →*demolish*→*shrines*⟩ into high level generalized concepts and relations, such as ⟨{*ISIS, mujahidin*}→{*kill, demolish*}→{*kafir, shrines*}⟩ by discovering contextual synonyms such as {*ISIS, mujahidin*}, {*kafir, shrines*} and {*kill, demolish*}. Note that contextual synonyms are not synonyms in the traditional dictionary sense, but they are phrases that may occur in similar semantic roles and associated with similar contexts.

$$\left.\begin{array}{c} \text{mujahidin} \rightarrow \text{kill} \rightarrow \text{kafir} \\ \\ \text{and} \\ \\ \text{ISIS} \rightarrow \text{demolish} \rightarrow \text{shrines} \end{array}\right\} \begin{array}{c} \{\text{ISIS, mujahidin}\} \\ \downarrow \\ \{\text{kill, demolish}\} \\ \downarrow \\ \{\text{kafir, shrines}\} \end{array}$$

Triplets extracted with semantic role labeling are noisy and sparse. We develop a hierarchical bottom-up merging algorithm that generalizes triplets into meaningful high level relationships. We achieve this by employing syntactic and semantic corpus-based criteria. *Syntactic criteria* are developed to merge a pair of subjects-verbs-objects only if they share common context related to their different arguments (i.e. a pair of different subjects are merged only if they co-occur with an identical verbs-objects context). The details of the syntactic criteria are presented in Section 5.1. Furthermore, a corpus-based *semantic criterion* is developed

ASONAM '15, August 25-28, 2015, Paris, France
© 2015 ACM. ISBN 978-1-4503-3854-7/15/08 $15.00
DOI: http://dx.doi.org/10.1145/2808797.2809312

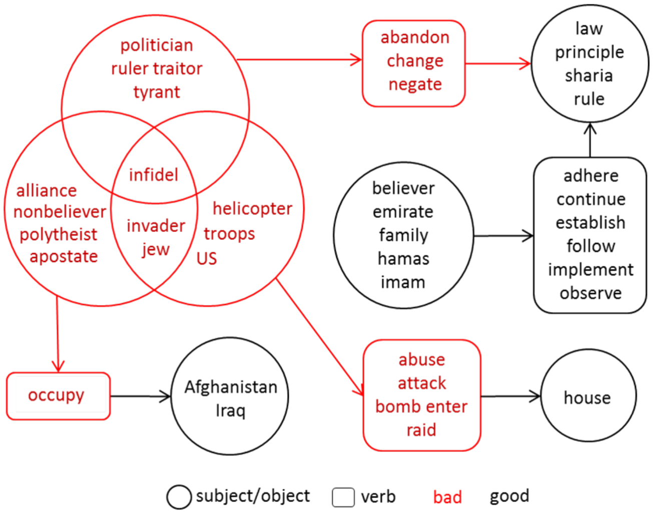


Figure 1. A sample semantic network learned from stories

for subjects, verbs and objects based on their shared verb-object, subject-object and subject-verb contexts correspondingly. The details of the semantic criterion are presented in Section 5.2. A hierarchical bottom-up merging algorithm, similar to the one employed in [2], allows information to propagate between clusters of relations and clusters of objects and subjects as they are created. Each cluster represents a high-level relation or concept. A concept cluster can be viewed as a node in a graph, and a relation cluster can be viewed as a link between the concept clusters that it relates.

Our proposed algorithm utilizes both syntactic and semantic corpus-based merging criteria. A pair of ⟨Subject, Verb, Object⟩ triplets is merged only if (i) they share a common context among their corresponding terms (i.e. syntactic criteria) and (ii) they satisfy corpus-based support and similarity measure thresholds (i.e. semantic criteria). A corpus-based measure of "contextual synonymy" will be defined based on their shared contexts of subjects, verbs and objects. We observed that this combination of criteria helps to generalize triplets into meaningful high-level concepts without *drift*. For example, Table 1 shows top ten contextual synonyms identified for three keywords selected from our extremist discourse corpus.

Generalized concept and relation clusters define a semantic network [3]. Collections of co-related contextual synonyms can be used to construct meta-nodes and links in a network describing the semantic space of the underlying texts. Components of the graph reveal networks of generalized concepts expressed as different groups of actors (subjects) performing various sets of actions (verbs) on different groups of targets/victims (objects). A sample network extracted from stories that mention Afghanistan and Iraq is shown in Figure 1. This technique contributes to the detection of narratives used by extremist groups to convey their ideology.

We evaluate the utility of generalized concepts by comparing their predictive accuracy when used as features in a story detection task. We aim to to develop a story classifier that can discriminate between stories and non-stories. A story is defined as an actor(s) taking action(s) that culminates in a resolution(s), e.g. *"Mujahedeen Imarah Islam Afghanistan attacked a military base in Hisarak district of Nangarhar province with heavy weapons on Tuesday. Reports indicate about 22 mortars landed on the base and causing a fatal loss enemy side."* A non-story paragraph is one in which there is no explicit resolution but hypothetical ones, e.g. *"Praised be God. We praise Him and seek His help and forgiveness. God save us from our evils and bad deeds. Whoever is guided by God, no one can mislead him, and whoever deviates no one can guide him."*

We use a corpus of $39,642$ paragraphs where $9,058$ paragraphs coded as stories, and $30,584$ paragraphs coded as non-stories by domain experts. We experiment with (i) standard keyword-based features, (ii) triplet-based features which generate sets of subjects and sets of objects associated with distinct verbs as features [4], and (iii) generalized concept/relation based features developed in this paper. Previously in [4], we obtained a precision of 73%, recall of 56% and F-measure of 63% for the detection of minority class (i.e. stories) by using triplet-based features, which provided a 161% boost in recall, and an overall 90% boost in F-measure over keyword-based features. In this paper, we show that when we utilize generalized concepts/relations extracted from the entire corpus of stories and non-stories as features, we obtain new highs in story detection accuracies as 86% precision, 82% recall and 85% F-measure. Generalized concepts/relations as features yield a 50% boost in recall at higher precision, and an overall 36% boost in story detection accuracy over verb-based triplet features developed earlier.

The contributions of this paper are: (i) a generalized concept/relationship representation of text that overcomes surface level differences (which arise when different keywords are used for related concepts) without drift (ii) a higher-level semantic network representation of related stories, and (iii) a 36% boost in the challenging automated story detection [5] task.

2. Related Work

Our paper has contributions in two distinct areas; unsupervised relation extraction and story detection. We present related work in these two areas separately.

2.1. Unsupervised Relation Extraction

Unsupervised learning of concepts and relations has become very popular in the last decade. One of the pioneering studies in the field, by Hasegawa et al. [6], clusters pairs of named entities according to the similarity of context words (predicates) in between. Each cluster represents a relation, and a pair of objects can appear in at most one cluster (relation). Our framework does not depend on the use of a Named Entity Recognition (NER) system and it allows subject and objects to appear in more than one relation.

Bank et al. [7] build soft clusters of named entities however their system require an external knowledge-base/ontology of relations to operate.

Kok and Domingos presented a similar framework to ours in their 2008 paper [2], which extracts concepts and relations together from ground facts also learning a semantic network. They use a purely statistical model based on second order Markov Logic and report performance in comparison with other clustering algorithms based on a manually created gold-standard. Our evaluation strategy compares the efficacy of concepts/relations as features with other feature sets on story detection task.

Recently, the focus of unsupervised information extraction has moved on to large web data sets creating the need for more scalable approaches. Kang et al. [1] deals with this problem using a parallel version of tensor decomposition. They learn contextual synonyms and generalized concepts/relations simultaneously, however they do not present any formal evaluation of their concepts/relations.

Another problem of dealing with web-scale discovery is polysemy and synonymy of verbs. Polysemy becomes a problem when the two occurrences of the same word which have different meanings are placed into the same cluster. Min et al. [8] addresses this problem by incorporating various semantic resources such as hyponymy relations, coordination patterns, and HTML structures. They observe that the best performance is achieved when various resources are combined together. We address word sense disambiguation by incorporating features from words' context. The contextual information flow via alternating merging of nouns and verbs handles the problems due to polysemy.

2.2. Story Detection

We study the problem of predicting whether or not a given paragraph tells a story. A story can be defined as "a sequence of events leading to a resolution or projected resolution". We perform supervised learning using a training set of stories and non-stories annotated by domain experts. Gordon et al. has published related work about story detection in conversational speech [9] and weblogs [10]. They use a confidence-weighted linear classifier with a variety of lexical features to classify weblog posts in the ICWSM 2009 Spinn3r Dataset and obtained the best performance [11] using unigram features with precision 66%, recall = 48%, F-score = 55%.

3. System Architecture

The main components of our system architecture are shown in Figure 2. The numbers on the top left corner of each box represent the order in which these processes are executed. Each process is briefly described below, while the details are presented in following sections.

1) Paragraphs in our dataset are annotated by human experts as Story and Non-Story. We treat each paragraph as a single data item to be classified.
2) Paragraphs are loaded into a SRL component. First, we apply co-reference resolution. Then, we use a shallow NLP parser and a post-processing step on the parse-tree in order to obtain the semantic role labels for $\langle Subject, Verb, Object \rangle$ triplets found in sentences. (See section 3.1).
3) Using the triplets, we create three separate pairwise contextual similarity matrices for subjects, verbs and objects based on their co-occurrences with verb-object, subject-object, and subject-verb pairs respectively. (See section 4).
4) Triplets and contextual similarity matrices are used as inputs to our clustering engine, which selectively merges and grows combined clusters of related subjects, verbs and objects. (See section 5).
5) Step 4 yields a number of concept (i.e. subject/object) clusters linked by relation (i.e. verb) clusters. (See section 5).
6) We further experiment with expanding these concepts and generalized relations with word-sense disambiguated dictionary look-ups in WordNet [12]. (See section 6.2).
7) We further expand these concepts and relations with word-sense disambiguated dictionary look-ups. (See section 6.2).
8) Both original and expanded concepts/relations are tested as features for the story/non-story classification task using ten-fold cross validation. (See section 6.4).

3.1. Semantic Role Labeler

We use the Stanford Deterministic Coreference Resolution System [13], [14], [15], [16] as a pre-processing step. Next, each paragraph is processed by ClearNLP shallow parser [17], which assigns a semantic role label to each word in a sentence. There are more than 40 possible labels[1] provided by ClearNLP. Currently, we are only interested in subjects, predicates and objects. In the final step, we apply post-processing on the output of ClearNLP to handle complex sentences with multiple verbs or some considerations for active or passive voice in order to extract related subject-verb-object triplets expressed in the sentence. Our framework can be adapted for languages other than English, provided that a semantic role labeler exists for that language.

4. Contextual Synonyms

We observe that a meaningful measure of pairwise similarity for subjects, verbs and objects can be obtained based on their shared verb-object, subject-object and subject-verb contexts, respectively. Therefore, we adapted the standard *bag-of-words* approach [18] to be used with triplets rather than regular text. For example, the similarity between a

[1] https://github.com/clir/clearnlp-guidelines/blob/master/md/dependency/dependency_guidelines.md

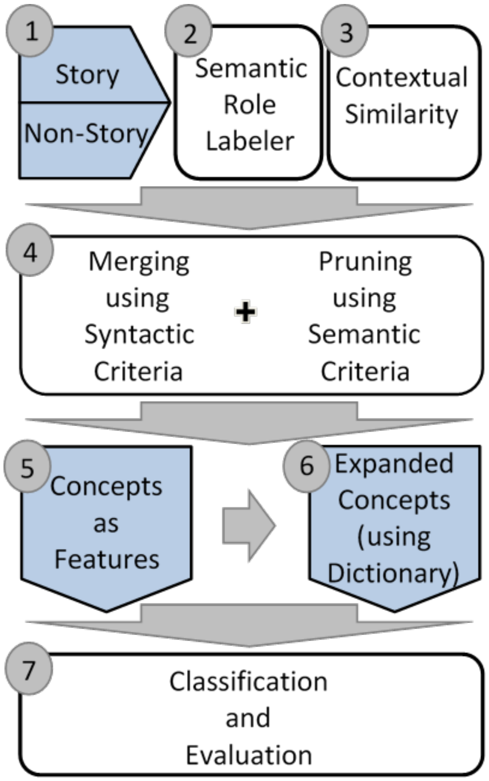


Figure 2. System Architecture

TABLE 1. TOP TEN CONTEXTUAL SYNONYMS FOR *mujahedeen*, *attack* AND *base*

mujahedeen	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

pair of subjects is determined by the frequency of their co-occurrences with the same verb-object pairs. In our preliminary experiments, we applied various clustering algorithms comparing different similarity measures such as euclidean and cosine however the contextual similarity measure defined in Figure 4 provides the most meaningful results. For example, in Table 1, *lion* is indeed among the most similar words for *mujahedeen* based on the contextual similarity measure, whereas none of the other standard similarity measures are able to retrieve this keyword.

```
1:  FIND CONCEPTS W/ UNIQUE PAIRS($\mathcal{T}, \mathcal{C}^0$)
2:      $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \leftarrow \emptyset$
3:      for all $\langle s_i, v_j, o_k \rangle \in \mathcal{T}$ do
4:          Find and add unique pairs to:
            $\mathcal{X} \leftarrow \mathcal{X} \cup \{\langle s_i, v_j \rangle\}$
            $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{\langle v_j, o_k \rangle\}$
            $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{\langle s_i, o_k \rangle\}$
5:      end for
6:      for all $\langle s_i, v_j \rangle \in \mathcal{X}$ and $\langle s_i, v_j, o_k \rangle \in \mathcal{T}$ do
7:          $\mathcal{C}^0 \leftarrow \mathcal{C}^0 \cup \{\langle s_i, v_j, O \rangle\}$ where $o_k \in O$.
8:      end for
9:      for all $\langle v_j, o_k \rangle \in \mathcal{Y}$ and $\langle s_i, v_j, o_k \rangle \in \mathcal{T}$ do
10:         $\mathcal{C}^0 \leftarrow \mathcal{C}^0 \cup \{\langle S, v_j, o_k \rangle\}$ where $s_i \in S$.
11:     end for
12:     for all $\langle s_i, o_k \rangle \in \mathcal{Z}$ and $\langle s_i, v_j, o_k \rangle \in \mathcal{T}$ do
13:         $\mathcal{C}^0 \leftarrow \mathcal{C}^0 \cup \{\langle s_i, V, o_k \rangle\}$ where $v_j \in V$.
14:     end for
15: end
```

Figure 3. Algorithm: Find concepts with unique pairs

Let $\mathcal{S}, \mathcal{V}$ and $\mathcal{O}$ be the set of all unique subjects, verbs and objects in our data set, respectively. And let $\mathcal{T}$ be the set of all $\langle s, v, o \rangle$ extracted triplets from our corpus, where $s \in \mathcal{S}$, $v \in \mathcal{V}$, $o \in \mathcal{O}$ denote a single subject, verb and object respectively. We calculate pairwise contextual similarity matrices ($S_{\mathcal{S}}$) for subjects, ($S_{\mathcal{V}}$) for verbs and ($S_{\mathcal{O}}$) for objects using the algorithms described in Figures 3 and 4. Throughout the rest of this paper, we refer to both noun and verb clusters as 'concepts' for simplicity.

Initially, we create concepts comprised of distinct pairs of subjects, verbs or objects with common context. We will name this initial set of concepts $\mathcal{C}^0$ in order to avoid confusion with the resulting set of concepts. Set $\mathcal{C}^0$ is composed of concepts c, each of which has a set of subjects (S), verbs (V) and objects (O), which co-occur with unique $\langle$verb, object$\rangle$, $\langle$subject,object$\rangle$ and $\langle$subject,verb$\rangle$ pairs respectively. The pseudo-code given in Figure 3 describes this procedure. In the first *for-loop* (lines 3–5, we iterate over all the $\langle s, v, o \rangle$ triplets and create a list of unique $\langle$subject,verb$\rangle$, $\langle$verb,object$\rangle$ and $\langle$subject,object$\rangle$ pairs. In the subsequent three *for-loops*, we grow our concept set at each iteration by adding a unique pair along with a set of all co-occurring words. Lines 6–8 perform this operation for $\langle$subject,verb$\rangle$ pairs, lines 9–11 for $\langle$verb,object$\rangle$ pairs and lines 12–14 for $\langle$subject,object$\rangle$ pairs.

After producing concepts with unique pairs, we proceed to calculate pairwise contextual similarity for subjects, verbs and objects. Let $n_s = |\mathcal{S}|$, $n_v = |\mathcal{V}|$ and $n_o = |\mathcal{O}|$ be the number of all unique subjects, verbs and objects in our corpus, respectively. We create similarity matrices $S_{\mathcal{S}} \in \mathbb{R}^{n_s \times n_s}$ for subjects, $S_{\mathcal{V}} \in \mathbb{R}^{n_v \times n_v}$ for verbs, and $S_{\mathcal{O}} \in \mathbb{R}^{n_o \times n_o}$ for objects. The algorithm in Figure 4 is used to fill in the similarity matrices. The similarity between a pair of words is defined as the number of common co-occurring unique contexts, i.e. if any of the two subjects, verbs or objects appear with the same verb-object, subject-

```
 1: CALCULATE CONTEXTUAL SIMILARITY($\mathcal{C}^0$)
 2:     $S_\mathcal{S}, S_\mathcal{V}, S_\mathcal{O} \leftarrow 0$
 3:     for all $c \in \mathcal{C}^0$ do
 4:         if $c = \langle S, v, o \rangle$ then
 5:             $S_\mathcal{S}(i,j) \leftarrow S_\mathcal{S}(i,j) + 1, \forall s_i, s_j \in S.$
 6:         else if $c = \langle s, V, o \rangle$ then
 7:             $S_\mathcal{V}(i,j) \leftarrow S_\mathcal{V}(i,j) + 1, \forall v_i, v_j \in V.$
 8:         else if $c = \langle s, v, O \rangle$ then
 9:             $S_\mathcal{O}(i,j) \leftarrow S_\mathcal{O}(i,j) + 1, \forall o_i, o_j \in O.$
10:         end if
11:     end for
12: end
```

Figure 4. Algorithm: Calculate contextual similarity

object or subject-verb pair respectively, then we increase the similarity count between those two words by one. In Figure 4, lines 4–5 calculate pairwise similarities between subjects, lines 6–7 for verbs and lines 8–9 for objects.

5. Concept and Relation Clustering

We follow a bottom-up agglomerative merging approach in order to populate our noun and verb clusters. The pseudocode for the algorithm is as shown in Figure 5. We start with the initial concept set, $\mathcal{C}^0$, that we created in Figure 3 and iteratively expand each element. First, each element of $\mathcal{C}^0$ is compared with the rest in order to create a set of candidates for merging based on the syntactic criteria (lines 5–6) described in Section 6.1. Next, we process each candidate and eliminate the words which fail the semantic criteria (lines 9–10) described in Section 6.2. We grow our candidate concepts by adding the elements which pass both tests (line 12). The main *while-loop*, beginning at line 3, continues to iterate until there are no more candidates suitable for merging. We explain the details of these syntactic and semantic criteria in the following two sections.

5.1. Syntactic Criteria

One of the major challenges in obtaining information via generalization is to maintain meaningful concepts as they grow. We address this problem by merging concepts only if they have a common context in all three semantic arguments (i.e. subject, verb, object). Given a generalized concept, $\langle \{s_1, s_2, ...\}, \{v_1, v_2, ...\}, \{o_1, o_2, ...\} \rangle \in \mathcal{C}$, we maintain that all subjects (s_i), verbs (v_j) and objects (o_k) are " contextually synonymous" among themselves and can be used interchangeably to generate meaningful triplets. Let $c_1 = \langle \{s_1, s_2\}, v_1, o_1 \rangle$ and $c_2 = \langle s_1, v_1, \{o_1, o_2\} \rangle$ be two concepts with unique pairs, i.e. $c_1, c_2 \in \mathcal{C}^0$. Consider merging these concepts into a more generalized concept $c_3 = \langle \{s_1, s_2\}, v_1, \{o_1, o_2\} \rangle$. Since c_3 adds a new object, o_2, to c_1, we require that c_1 and c_2 have a common context in order to justify the merge, i.e. the intersection of c_1 and c_2's subject and verb sets, $\{s_1\}$ and $\{v_1\}$, should be non-empty. Similarly, since we are adding a new subject, s_2, to c_2's subject set, we also require that the intersection of c_1 and c_2's verb and object sets, $\{v_1\}$ and $\{o_1\}$, should be non-empty as well. Since these conditions are satisfied in this case, we can merge c_1 and c_2 into the same concept provided they satisfy the semantic criteria discussed in the next section.

```
 1: CLUSTER CONCEPTS($\mathcal{T}, S_\mathcal{S}, S_\mathcal{V}, S_\mathcal{O}, \mathcal{C}^0$)
 2: $\mathcal{C} \leftarrow \mathcal{C}^0$
 3: while $flag = 1$ do
 4:   $flag \leftarrow 0$
 5:   for all $c \in \mathcal{C}^0$ do
 6:     Find matching concepts $\mathcal{M}$ using Syntactic Criteria
 7:     if $|\mathcal{M}| \geq 1$ then
 8:       $flag \leftarrow 1$
 9:       for all $m \in \mathcal{M}$ do
10:         $\{c\} \leftarrow \{c\} \cup \{m\}$
11:         Prune $c$ using Semantic Criteria.
12:         $\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$
13:       end for
14:     end if
15:   end for
16: end while
17: end
```

Figure 5. Bottom-Up Agglomerative Clustering Algorithm

On the other hand, let us consider $c_1 = \langle \{s_1, s_2\}, v_1, o_1 \rangle$ and $c_2 = \langle s_1, \{v_1, v_2\}, o_2 \rangle$. If we merge these concepts, the new concept will be $c_3 = \langle \{s_1, s_2\}, \{v_1, v_2\}, \{o_1, o_2\} \rangle$. Since c_3 adds a new object, o_2, to c_1, we require that the intersection of c_1 and c_2's subject and verb sets, $\{s_1\}$ and $\{v_1\}$, should be non-empty, which is the case. c_3 would also add a new verb, v_2, to c_1, hence we require that the intersection of c_1 and c_2's subject and object sets should be non-empty as well, which is not the case. There is a common subject but objects are totally distinct. Therefore we should not merge these concepts into the same one since there is not enough common context to justify the merged concept. We express these conditions in a more formal way, as follows.

Let $\mathcal{C}_1 = \langle \mathcal{S}_1, \mathcal{V}_1, \mathcal{O}_1 \rangle$ and $\mathcal{C}_2 = \langle \mathcal{S}_2, \mathcal{V}_2, \mathcal{O}_2 \rangle$ be two concepts. We merge $\mathcal{C}_1$ and $\mathcal{C}_2$ if they meet all of the following conditions:

- $\mathcal{S}_1 \neq \mathcal{S}_2 \Rightarrow \{\mathcal{V}_1 \cap \mathcal{V}_2 \neq \emptyset$ and $\mathcal{O}_1 \cap \mathcal{O}_2 \neq \emptyset\}$
- $\mathcal{V}_1 \neq \mathcal{V}_2 \Rightarrow \{\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$ and $\mathcal{O}_1 \cap \mathcal{O}_2 \neq \emptyset\}$
- $\mathcal{O}_1 \neq \mathcal{O}_2 \Rightarrow \{\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$ and $\mathcal{V}_1 \cap \mathcal{V}_2 \neq \emptyset\}$

5.2. Semantic Criteria

While the syntactic criteria ensure inter-relatedness of distinct members of concepts to their contexts, we also utilize a secondary measure to establish intra-relatedness between the distinct members of concepts in each argument position. We utilize the contextual similarity measure (defined in Figure 4) that relates subjects, verbs, and objects among themselves. The semantic test requires that only the most similar candidate keywords can be added to a concept. We use these criteria to grow the concepts without drift. We formally present semantic criteria as follows.

Let $\mathcal{C}_1 = \langle \mathcal{S}_1, \mathcal{V}_1, \mathcal{O}_1 \rangle$ and $\mathcal{C}_2 = \langle \mathcal{S}_2, \mathcal{V}_2, \mathcal{O}_2 \rangle$ be two concepts which passes the syntactic criteria and let $\mathcal{C}_3$ be the new concept after merging. Semantic criteria are applied as follows:

- Define $\mathcal{S}_{\text{int}} = \mathcal{S}_1 \cap \mathcal{S}_2, \mathcal{V}_{\text{int}} = \mathcal{V}_1 \cap \mathcal{V}_2, \mathcal{O}_{\text{int}} = \mathcal{O}_1 \cap \mathcal{O}_2$.
- Define
$$\begin{aligned}\mathcal{S}_{\text{diff}} &= (\mathcal{S}_1 \setminus \mathcal{S}_2) \cup (\mathcal{S}_2 \setminus \mathcal{S}_1)\\ \mathcal{V}_{\text{diff}} &= (\mathcal{V}_1 \setminus \mathcal{V}_2) \cup (\mathcal{V}_2 \setminus \mathcal{V}_1)\\ \mathcal{O}_{\text{diff}} &= (\mathcal{O}_1 \setminus \mathcal{O}_2) \cup (\mathcal{O}_2 \setminus \mathcal{O}_1)\end{aligned}$$
- Define $\mathcal{S}^*_{\text{int}}, \mathcal{V}^*_{\text{int}}$ and $\mathcal{O}^*_{\text{int}}$ to be the sets composed of the closest contextual synonyms of all words in $\mathcal{S}_{\text{int}}, \mathcal{V}_{\text{int}}$ and $\mathcal{O}_{\text{int}}$, respectively. In this step, we use the contextual similarity metric from the algorithm presented in Figure 4.
- Initially, $\mathcal{C}_3$ contains only the intersections of $\mathcal{C}_1$ and $\mathcal{C}_2$, i.e. $\mathcal{C}_3 = \langle \mathcal{S}_{\text{int}}, \mathcal{V}_{\text{int}}, \mathcal{O}_{\text{int}} \rangle$. We grow $\mathcal{C}_3$ by adding words from the difference sets of $\mathcal{C}_1$ and $\mathcal{C}_2$ only if they are among the closest contextual synonyms of the words in the intersections. Formally,
$$\begin{aligned}\mathcal{C}_3 = \langle\ & (\mathcal{S}_{\text{diff}} \cap \mathcal{S}^*_{\text{int}}) \cup (\mathcal{S}_1 \cap \mathcal{S}_2),\\ & (\mathcal{V}_{\text{diff}} \cap \mathcal{V}^*_{\text{int}}) \cup (\mathcal{V}_1 \cap \mathcal{V}_2),\\ & (\mathcal{O}_{\text{diff}} \cap \mathcal{O}^*_{\text{int}}) \cup (\mathcal{O}_1 \cap \mathcal{O}_2)\ \rangle.\end{aligned}$$

6. Experimental Evaluation

6.1. Data Set

We use a corpus of $39,642$ paragraphs where $9,058$ are coded as stories and $30,584$ as non-stories by domain experts. Text is collected from websites, blogs and other news sources that are known to be outlets of extremist groups such as Al-Qaeda, ISIS or their followers who sympathize with their cause and methods.

6.2. Expansion of Concepts with Dictionary-based Synonyms

After the Bottom-Up Agglomerative Clustering procedure (in Figure 5) terminates, we obtain high-level concepts and relations, which we refer to as ‘Tier 1’. In order to expand the concepts further with keywords that are missing in the training corpus, we experiment with adding sense-disambiguated WordNet [12] synonyms to ‘Tier 1’ obtaining ‘Tier 1 + WordNet’. Alternatively, we also utilize contextual similarity index to create ‘Tier 1 + Similarity’ as follows. For each concept $c = \langle \{s_{1\ldots m}\}, \{v_{1\ldots n}\}, \{o_{1\ldots l}\} \rangle \in \mathcal{C}$, where $m, n, l > 1$, we create a set of candidates to merge by picking the synonyms of each subject, $(s_i, 1 \leq i \leq m)$, verb $(v_j, 1 \leq j \leq n)$ and object $(o_k, 1 \leq k \leq l)$. Without loss of generality, we add w, synonym of s_i to cluster c, only if there is at least one triplet in our database, $\langle s, v, o \rangle \in \mathcal{T}$ such that $w = s$, $v \in \{v_{1\ldots n}\}$ and $o \in \{o_{1\ldots l}\}$.

6.3. Feature Matrix Generation

We report the results of story detection task using five different feature sets: (i) keywords, (ii) verb-based features extracted from triplets [4], (iii) concepts-based features (Tier 1) developed in this paper, (iv) concepts expanded with contextual similarity index (Tier 1 + Similarity) and (v) concepts expanded with WordNet (Tier 1 + WordNet).

6.3.1. Verb-based Features.

In our previous paper [4], we followed a standard verb-based approach to extract simple subject, object and preposition clauses associated with verbs found in story and non-story paragraphs. For each verb (V) mentioned in a story (S), and non-story (NS), we generated following set-valued features by using the training data:

- Argument list for S.V.Subjects, S.V.Objects, S.V.Prepositions for each verb V and story S.
- Argument list for NS.V.Subjects, NS.V.Objects, NS.V.Prepositions for each verb V and non-story NS.

For each test paragraph P, for each verb V in P, we extracted its typed argument lists P.V.Subjects, P.V.Objects and P.V.Prepositions. Then, we matched them to the argument lists of the same verb V. A match succeeds if the overlap between a feature's argument list (e.g. S.V.Subjects, or NS.V.Subjects) covers the majority of the test paragraph's corresponding verb argument list (e.g. P.V.Subjects).

6.3.2. Concepts-based Features.

In this paper, first, we generate the concepts for story and non-story paragraphs by using the training data. Next, we process each test paragraph P, and generate its semantic triplets, $\langle s, v, o \rangle$. A binary feature matrix is created by checking if any of the semantic triplets of P matches a concept, $\langle S, V, O \rangle$, where S, V and O are related sets of subjects, verbs and objects respectively. A match succeeds if $s \in S, v \in V$ and $o \in O$.

6.4. Cross Validation for Detecting Stories

We evaluate the quality of generalized concepts and relations by their performance as features in story detection. The goal is to improve the predictive accuracy of story/non-story classifier through the use of these new features. We experiment with several different supervised learning packages including SVM [19], decision trees [20] and SLEP [21] concluding that SLEP outperforms others for this task. We use the MATLAB implementation of SLEP package [22] and obtained the best results using LogisticR model. Training and testing are performed using ten-fold cross validation and repeated with random shuffling over multiple iterations. The results are averaged over all iterations. We report the predictive performance of SLEP classifier using various feature sets for story and non-story categories in Tables 2 and 3.

The feature sets we used are keywords, verb-based features [4] (Triplets), concepts and relations (Tier 1), concepts/relations expanded with contextual similarity index

TABLE 2. Performance of classifier for Stories

Method	Precision	Recall	F-Measure
Keywords	0.81	0.20	0.33
Triplets	0.73	0.55	0.62
Tier 1	**0.87**	0.78	0.83
Tier 1 + Similarity	0.86	**0.82**	**0.84**
Tier 1 + WordNet	0.87	0.80	0.83

(Tier 1 + Similarity) and concepts/relations expanded with WordNet (Tier 1 + WordNet). The feature sets produced by the bottom-up agglomerative clustering algorithm outperform others in the story detection category. We gained 7% boost in precision, 50% boost in recall and 36% boost in F-Measure over the best performance of keyword and triplet features (see Table 2). We observe that high-level concepts/relations have far more discriminative power compared to other features. A key reason is that they are able to eliminate dependent features by generalization. There is not a big difference in performance among the original and expanded concept-based feature sets and we can clearly see that WordNet expansion did not contribute to the performance of concepts expanded by contextual similarity. This finding presents another strong point in favor of our framework since adding information from an external knowledge-base was not able to provide a boost.

In the non-story category (Table 3), concept-based features are lagging behind in performance. This may be due to the structural diversity of non-story paragraphs since there are several different sub categories among them [23]. Another observation is that concept-based features help overcome the performance bias between story and non-story categories due to the imbalance in the number of training samples. Overall, concepts/relations deliver a 36% boost in performance for story detection.

6.5. Sensitivity Analysis

We assess concept/relation based features against the possibility of over-fitting since they are highly dependent on the training corpus. We explore this issue by using the regularization parameter, λ in SLEP's optimization formulation. We can pin-point the optimal number of features and avoid over-fitting by observing the performance of the system as the value of λ changes. The plots given in Figure 6 display the change in λ versus the number of features (middle row) and the performance (precision, recall and F-Measure) for story (top row) and non-story (bottom row) categories. In both cases, we can observe that there is a sharp drop in the number of features ($12,000$ to $2,000$) around $10^{-5} \leq \lambda \leq 10^{-4}$ while the precision, recall and F-Measure are preserved. The data cursor box in the middle plot mark the point of optimal value for λ and the corresponding number of features. Experimentally, we identified the optimal number of features as $7,563$ which prevents over-fitting of the model and preserves the gains in performance.

TABLE 3. Performance of classifier for Non-Stories

Method	Precision	Recall	F-Measure
Keywords	0.90	0.98	0.94
Triplets	0.89	0.99	0.92
Tier 1	0.80	**0.89**	0.84
Tier 1 + Similarity	0.83	0.86	0.84
Tier 1 + WordNet	0.82	**0.88**	**0.85**

7. Conclusion

We presented an algorithm for discovering generalized concept/relationship representation of a collection of related documents that overcomes surface level differences which arise when different keywords are used for related concepts. This representation provides a 36% boost in the challenging automated story detection task and a higher-level semantic network representation of related stories. In future work, we plan to gauge the utility of generalized concepts in document clustering tasks. We plan to use a bi-clustering approach which can point to subsets of stories and associated generalized concepts/relations as their themes. Since clustering is unsupervised, we need to rely on domain expert knowledge to evaluate the quality of the detected clusters and their themes. We also plan to develop visualization tools for exploring document collections and their clusterings through their high-level semantic network representations.

Acknowledgment

This research was supported by an Office of Naval Research grants N00014-09-1-0872 and N00014-14-1-0477 performed at Arizona State University. Some of the material presented here was sponsored by Department of Defense and is approved for public release, case number:15-467.

References

[1] U. Kang, E. Papalexakis, A. Harpale, and C. Faloutsos, "Gigatensor: Scaling tensor analysis up by 100 times - algorithms and discoveries," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 316–324.

[2] S. Kok and P. Domingos, "Extracting semantic networks from text via relational clustering," in *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, 2008, pp. 624–639.

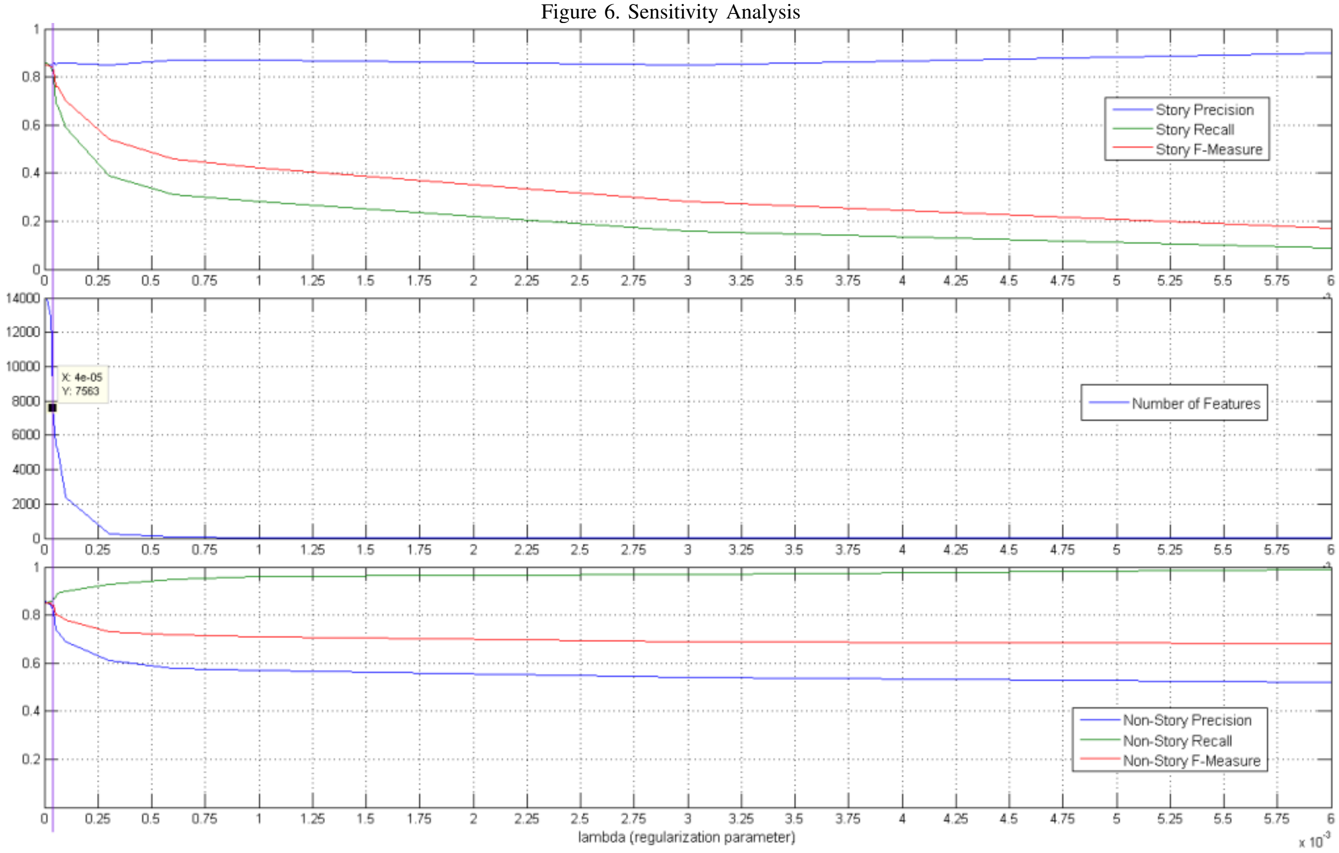
[3] M. R. Quillian, "Semantic memory," in *Semantic Information Processing*, 1968, pp. 227–270.

[4] B. Ceran, R. Karad, A. Mandvekar, S. R. Corman, and H. Davulcu, "A semantic triplet based story classifier," *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, vol. 0, pp. 573–580, 2012.

[5] J. Allan, V. Lavrenko, and H. Jin, "First story detection in tdt is hard," in *Proceedings of the Ninth International Conference on Information and Knowledge Management*, ser. CIKM '00, 2000, pp. 374–381.

[6] T. Hasegawa, S. Sekine, and R. Grishman, "Discovering relations among named entities from large corpora," in *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ser. ACL '04, 2004.

Figure 6. Sensitivity Analysis



[7] M. Banko and O. Etzioni, “Strategies for lifelong knowledge extraction from the web,” in *Proceedings of the 4th International Conference on Knowledge Capture*, ser. K-CAP ’07, 2007, pp. 95–102.

[8] B. Min, S. Shi, R. Grishman, and C.-Y. Lin, “Ensemble semantics for large-scale unsupervised relation extraction,” in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1027–1037.

[9] 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*, 2005, pp. 145–152.

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

[11] 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*, 2009.

[12] (2010) About wordnet. Princeton University. [Online]. Available: http://wordnet.princeton.edu

[13] K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning, “A multi-pass sieve for coreference resolution,” in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 492–501.

[14] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky, “Stanfords multi-pass sieve coreference resolution system at the conll-2011 shared task,” *CoNLL 2011*, p. 28, 2011.

[15] H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, and D. Jurafsky, “Deterministic coreference resolution based on entity-centric, precision-ranked rules,” *Comput. Linguist.*, vol. 39, no. 4, pp. 885–916, Dec. 2013.

[16] M. Recasens, M. C. de Marneffe, and C. Potts, “The life and death of discourse entities: Identifying singleton mentions,” in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 627–633.

[17] J. D. Choi, “Optimization of natural language processing components for robustness and scalability,” Ph.D. dissertation, University of Colorado at Boulder, 2012.

[18] N. Ide and J. Véronis, “Introduction to the special issue on word sense disambiguation: The state of the art,” *Comput. Linguist.*, vol. 24, pp. 2–40, 1998.

[19] C. Chang and C. Lin, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[20] (2013) Matlab statistics and machine learning toolbox. The MathWorks Inc. [Online]. Available: http://www.mathworks.com/help/stats/classification-trees-and-regression-trees-1.html

[21] J. Liu, J. Chen, and J. Ye, “Large-scale sparse logistic regression,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 547–556.

[22] J. Liu, S. Ji, and J. Ye. (2009) Slep: Sparse learning with efficient projections. Arizona State University. [Online]. Available: http://www.public.asu.edu/~jye02/Software/SLEP

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