

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/261088075>

A Semantic Triplet Based Story Classifier

Conference Paper · August 2012

DOI: 10.1109/ASONAM.2012.97

CITATIONS

7

READS

37

5 authors, including:



[Ravi Karad](#)

Arizona State University

1 PUBLICATION 7 CITATIONS

SEE PROFILE



[Ajay Mandvekar](#)

Arizona State University

1 PUBLICATION 7 CITATIONS

SEE PROFILE



[Steven R. Corman](#)

Arizona State University

66 PUBLICATIONS 1,256 CITATIONS

SEE PROFILE



[Hasan Davulcu](#)

Arizona State University

116 PUBLICATIONS 878 CITATIONS

SEE PROFILE

All content following this page was uploaded by [Steven R. Corman](#) on 27 August 2015.

The user has requested enhancement of the downloaded file. All in-text references [underlined in blue](#) are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

A Semantic Triplet Based Story Classifier

Betul Ceran*, [Ravi Karad*](#), [Ajay Mandvekar*](#), [Steven R. Corman†](#) and [Hasan Davulcu*](#)

**School of Computing, Informatics and Decision Systems Engineering
Arizona State University, Tempe, AZ 85287-8809.*

Email: {betul, rkarad, amandvek, hdavulcu}@asu.edu

†*Hugh Downs School of Human Communication,
Arizona State University, Tempe, AZ 85287-1205.*

Email: steve.corman@asu.edu

Abstract

A story is defined as “an actor(s) taking action(s) that culminates in a resolution(s).” In this paper, we investigate the utility of standard keyword based features, statistical features based on shallow-parsing (such as density of POS tags and named entities), and a new set of semantic features to develop a story classifier. This classifier is trained to identify a paragraph as a “story,” if the paragraph contains mostly story(ies). Training data is a collection of expert-coded story and non-story paragraphs from RSS feeds from a list of extremist web sites. Our proposed semantic features are based on suitable aggregation and generalization of <Subject, Verb, Object> triplets that can be extracted using a parser. Experimental results show that a model of statistical features alongside memory-based semantic linguistic features achieves the best accuracy with a Support Vector Machine (SVM) classifier.

1. Introduction

Personal narratives are powerful sources of persuasion, none more so than stories that cultural heroes tell about their own lives [1]. 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 [2]. 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 serves 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 of developing an automated story classifier is to reduce 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, everyday people and so on. Second, the actors must be performing actions. This can include fighting, preparing for a battle, talking to others and so on. Third, the actors actions must result in a resolution. Resolutions can include a new state of affairs, a new equilibrium created, a previous equilibrium restored, victory and so on. Besides, stories usually have story worlds, or worlds where 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, 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, Verse or Other. 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.

In this paper, we utilize a corpus of 16,930 paragraphs where 3,301 paragraphs coded as stories, and 13,629 paragraphs coded as non-stories by domain experts to develop a story classifier. Training data is a collection of Islamist extremist texts, speeches, video transcripts, forum posts, etc., collected in open source. We investigate the utility of standard keyword based features, statistical features that can be extracted using shallow-parsing (such as density of POS tags and density of named entities), and a new set of semantic features in development of a story classifier. Our study is motivated by the observation [3] 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 contribution of this paper is the introduction of a new set of semantic features based on related linguistic subject, verb, object categories that we named as *triplet based verb features* which are motivated by the definition of “story” as “actors taking actions that culminate in resolutions.”. Our proposed semantic features are based on suitable aggregation and generalization of <Subject, Verb, Object> triplets that can be extracted using a shallow-parser. Experimental results (see Table 4) show that a combination of statistical part-of-speech (POS) and named-entity (NE) features, with semantic triplet-based features achieves highest accuracy with a Support Vector Machine (SVM) based classifier. We obtained precision of 73%, recall of 56% and F-measure of 0.63 for minority class (i.e. stories) which indicates a **161%** boost in recall, and an overall **90%** boost in F-measure with negligible reduction in precision through the utility of triplet based features over standard keyword based features.

2. Related Work

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 the ICWSM 2009 Spinn3r Dataset, and they obtained 66% precision, 48% recall, and F-measure of 0.55.

3. System Architecture

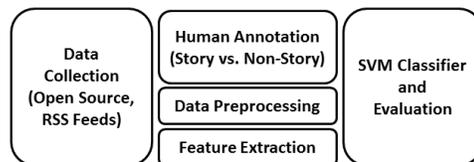


Figure 1. System Architecture

3.1. Data Collection

Our corpus is comprised of 16,930 paragraphs from extremist texts collected in open source. Stories were drawn from a database of Islamist extremist texts. Texts were selected by subject matter experts who consulted open source materials, including opensource.gov, private collection/dissemination groups, and known Islamist extremist web sites and forums. Texts come from groups including al-Qaeda, its affiliates, and groups known to sympathize with its cause and methods. The subject matter experts selected texts which they believe contained or were likely to contain stories, defined as a sequence of related events, leading to a resolution or projected resolution.

Extremists’ texts are rarely, if ever, composed of 100% stories, and indeed the purpose of this project is to enable the detection of portions of texts that are stories. Accordingly, we developed a coding system consisting of eight mutually-exclusive and exhaustive categories: story, exposition, imperative, question, supplication, verse, annotation, and other along with

definitions and examples on which coders could be trained. After training, coders achieved reliability of Cohen’s Kappa = 0.824 (average across eleven randomly sampled texts). Once reliability of the coders and process was established, single coders coded the remainder of the texts, with spot-check double coding to ensure reliability was maintained.

The Cohen’s Kappa measure represents how two observers agree on sorting items into different categories. The observers can be human or machine. The range of Cohen’s Kappa varies between 0 and 1. Fleiss [10] characterizes Kappa range over 0.75 as excellent, range between 0.40 to 0.75 as fair to good, and less than 0.40 as poor. Hence coders’ reliability of 0.824 falls into the range of excellent. After training and testing with ten-fold cross-validation, we calculated the agreement between classifier algorithm and the human coders as Cohen’s Kappa = 0.48, which falls into the range of fair to good.

3.2. Data Preprocessing

3.2.1. Named Entity Recognition Tagger. Named entity recognition (NER) [11] (also known as entity identification or entity extraction) is a subtask of information extraction that seeks to locate and classify atomic elements in text into predefined categories such as persons, organizations, locations. Research indicates that even state-of-the-art NER systems are brittle, meaning that NER systems developed for one domain do not typically perform well on other domains [12]. For the purpose of annotating the entities found within the texts belonging to extremist narratives, we used the most popular publicly available NER libraries. We evaluated three libraries: Stanford Named Entity Recognizer [13], Illinois Named Entity Tagger [14] and an online web service provided by OpenCalais [15].

A set of six documents belonging to extremist narratives were manually annotated by a specialist as a person, a location or an organization. Next, the same set of documents was annotated using NER libraries in order to determine their accuracy. F-measure was used to measure their accuracy as shown in Table 1.

We also developed a consensus analysis based algorithm, which we named as ‘democratic NER tagger’, using the output of all three taggers as follows:

- 1) For a particular text document to be annotated for named-entity tags, invoke each NER tagger (Stanford, Illinois and Open Calais).
- 2) For an annotated entity/word determine the category (Person, Organization, Location) assigned by each NER tagger.

Table 1. NER Tagger F-measures

Text-ID	DNERT	Stanford	Illinois	Open Calais
1	0.592	0.355	0.463	0.312
2	0.567	0.587	0.549	0.164
3	0.652	0.627	0.574	0.247
4	0.837	0.867	0.867	0.591
5	0.720	0.686	0.483	0.459
6	0.505	0.446	0.651	0.416
Average	0.644	0.594	0.597	0.364

- 3) If the phrase is classified by only one tagger, then assign it as the final tag.
- 4) If all taggers disagree on the category of a phrase, then we pick the final category according to the accuracy of the taggers as follows:
 - Illinois NER has the highest accuracy for Locations and Organizations;
 - Stanford NER has the highest accuracy for Persons.
- 5) If two out of three NER taggers agree on the predicted category of a phrase, then the final category is determined by majority agreement.

Within the six documents, specialist annotated 308 organization names, 259 location names, and 127 person names. Table 1 summarizes the accuracies of the software libraries, as well as the accuracy of our simple democratic NER tagger (DNERT) which relies on all of them. Overall, our democratic NERT achieves the highest performance compared to individual NER taggers.

3.2.2. Named Entity Standardization. The extremist texts under consideration are collected from various social media sources (i.e. blogs, organizations’ websites and RSS feeds). Since these are all human generated documents, they are rife with misspellings and aliases of named entities. For example, the person entity ‘Osama Bin Laden’ is sometimes referred to as ‘Bin Laden’, ‘Sheikh Osama’, or it is sometimes misspelled as ‘Osamma Bin Laden’ or ‘Usama’. In order to standardize the usage of the entities and improve the accuracy of classifier dependent on the entity features we have come up with a two step named entity standardization process:

- **Misspelling Correction Step**

The named entity spelling is corrected using a lookup with the Google’s ‘Did u mean?’ feature. This process enables us to identify the most correct spelling of a named entity. E.g. An entity named ‘Osamma bin laden’ is corrected as ‘Osama bin laden’.

- **Standardization for Aliases**

We query the RDF data stores of DBpedia[16]

Table 2. Named-Entity Correction and Standardization Results

# of Occurrences	Person	Organization	Location
Total	5015	2456	4279
Distinct	332	200	290
Standardized	72	26	30
Accuracy	65 (90.3%)	24 (92.3%)	28 (93.3%)

in order to find a standardized name for all location, organization and person entities. DBpedia data set has a public SPARQL endpoint available at: <http://DBpedia.org/sparql>. The DBpedia OWL ontology for named entity types have following properties where known aliases are stored: *dbpprop:alternativeNames*, *dbpprop:name*, *foaf:name*. By querying the alternative names we are able to obtain the standard name for each entity.

Table 2 summarizes the results for the named entity correction and standardization algorithm. First row shows the total number of occurrences of each type of named entity in our document corpus. Second row shows the counts of distinct entities by their type. Third row shows the number of distinct named entity corrections made through above spelling correction and alias standardization procedure. The changes were manually evaluated by a human annotator to verify their accuracy. As seen in the final row of the table, out of 72 person name changes, 65 were accurately standardized by the algorithm, providing 90% accuracy for person entities. Similarly, for organization entities the correction and standardization accuracy is around 92%, and for location entities it is around 93%.

3.3. Human Annotation: Story vs. Non-Story Coding

Stories are differentiated from non-stories as follows: Because they describe actions, stories will have a lower proportion of stative verbs than non-stories. Stories will include more named entities, especially person names, than non-stories. Stories will use more personal pronouns than non-stories. Stories may include more past tense verbs (*i.e.*, X resulted in Y, X succeeded in doing Y, etc.) than non-stories. Stories may repeat similar nouns. For example, “mujahedeen” may be mentioned in the beginning of the story and then again at the end of the story. Paragraphs with stories in them have different sentence lengths than paragraphs without stories in them.

3.4. Feature Extraction

Above observations made in Section 3.3 motivates following standard keyword based features, statistical features based on shallow-parsing, and a new set of semantic features for the development of a story classifier:

- **Keywords:** TF/IDF measure [17] is calculated for each word contained in the whole paragraph set. Then a certain number of terms, in our case 20,000, with the top TF/IDF values are selected as features. Then term-document frequency matrix is created out these keyword features.
- **Density of POS Tags:** Part of Speech (POS) Tag Ratios [18] for each document is calculated with respect to numbers of tokens.
- **Density of Named Entities:** Named Entity (NE) Tag frequency [13] per document is calculated. The tags are Person, Location and Organization.
- **Density of Stative Verbs:** Some other statistical features are also included in all experiments, such as the number of valid tokens and the ratio between observed stative verbs and total number of verbs in a paragraph.
- **Semantic Triplets Extraction:** We present our semantic triplet extraction methods in Section 3. We also discuss how triplets from stories and non-stories are aggregated and generalized to form memory-based features for verbs.

3.5. Support Vector Machine (SVM) Classifier

SVM [19] is a supervised learning technique which makes use of a hyperplane to separate the data into two categories. SVM is originally proposed as a linear classifier [20] but later improved by the use of kernel functions to detect nonlinear patterns underlying the data [21]. There are various types of kernel functions available [22]. In this study, we use RBF kernel defined as $K(x_i, x_j) = e^{-\|x_i - x_j\|}$, where $x_{i,j}$ are data points [23].

3.6. Training and Testing

The corpus contains 1,256 documents containing both story and non-story paragraphs. There are a total of 16,930 paragraphs, where 13,629 paragraphs classified reliably as non-stories, and 3,301 paragraphs classified as stories by domain experts. In our evaluations, we performed 10 fold cross validation with the document files as follows: we break documents into 10 sets of size $n/10$, where n is total number of documents

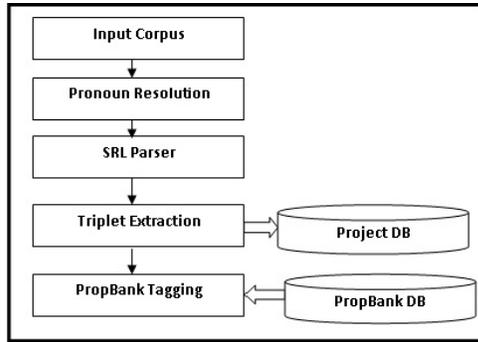


Figure 2. Triplet Extraction Pipeline

(1,256). During the training phase, both story and non-story paragraphs from 9/10 documents are used as the training set, their features are extracted, and a classifier is trained. During the testing phase, the remaining 1/10th of the documents are used; the features for both stories and non-stories are extracted, and matched to the features extracted during the training phase. Doing this evaluation, we are ensuring that training and test data features are in fact coming from different documents. We calculate precision, recall for each iteration of the 10 fold cross validation and we report mean precision, recall for both both stories and non-stories.

4. Semantic Triplet Extraction

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 [24], [25] is the process of finding who (subject), is doing what (verb) with/to whom (direct objects), when and where (indirect objects/and prepositions). Our triplet extraction utilizes the information extraction pipeline shown in Figure (2).

4.1. Pronoun Resolution

Interactions are often specified through pronominal references to entities in the discourse, or through co references where, a number of phrases are used to refer to the same entity. Hence, a complete approach to extracting information from text should also take into account the resolution of these references. Our pronoun

resolution module [26], [27] uses a heuristic approach to identify the noun phrases referred by the pronouns in a sentence. The heuristic is based on the number of the pronoun (singular or plural) and the proximity of the noun phrase. The closest earlier mentioned noun phrase that matches the number of the pronoun is considered as the referred phrase.

4.2. Semantic Role Labeler (SRL) Parser

SRL parser [28] is the key component of our triplet extractor. To extract the subject-predicate-object from an input sentence, important step is identifying these elements in a sentence and parse it. SRL parser does exactly the same. SRL is propriety software developed by Illinois research group and its shallow semantic parser. The goal of the semantic role labeling task is to discover the predicate-argument structure of each predicate that fill a semantic role and to determine their role (Agent, Patient, Instrument etc). As shown in the following example, SRL is robust in identifying verbs, and their arguments and argument types accurately in the presence of syntactic variations.

Numbered arguments (A0-A5, AA): Arguments define verb-specific roles. They depend on the verb in a sentence. The most frequent roles are A0 and A1 and, commonly, A0 stands for the agent and A1 corresponds to the patient or theme of the proposition.

Adjuncts (AM-): General arguments that any verb may take optionally. There are 13 types of adjuncts: AM-ADV - general-purpose, AM-MOD - modal verb, AM-CAU - cause, AM-NEG - negation marker, AM-DIR - direction, AM-PNC - purpose, AM-DIS - discourse marker, AM-PRD - predication, AM-EXT - extent, AM-REC - reciprocal, AM-LOC - location, AM-TMP - temporal, AM-MNR - manner.

References (R-): Arguments representing arguments realized in other parts of the sentence. The label is an R- tag prefixed to the label of the referent, e.g. [A1 The pearls] [R-A1 which] [A0 I] [V left] [A2 to my daughter-in-law] are fake.

SRL System Architecture: SRL works in four-stages, starting with pruning of irrelevant arguments, identifying relevant arguments, classifying arguments and inference of global meaning.

Pruning - Used to filter out simple constituents that are very unlikely to be arguments.

Argument Identification - Utilizes binary classification to identify whether a candidate is an argument or not. The classifiers are applied on the output from the pruning stage. A simple heuristic is employed to filter out some candidates that are obviously not arguments.

Argument Classification - This stage assigns labels

to the argument candidates identified in the previous stage.

Inference - In the previous stages, decisions were always made for each argument independently, ignoring the global information across arguments in the final output. The purpose of the inference stage is to incorporate such information, including both linguistic and structural knowledge. This knowledge is useful to resolve any inconsistencies of argument classification in order to generate final legitimate predictions.

4.3. Triplet Extraction

Our triplet extraction algorithm processes SRL output. The SRL output has a specific multi-column format. Each column represents one verb (predicate) and its arguments (A0, A1, R-A1, A2, etc) potentially forming many triplets. For a simple sentence, we can read one column and extract a triplet. For complex sentences with many verbs, we developed a bottom-up extraction algorithm for detecting and tagging nested events. We will illustrate our approach using the following example.

Example Paragraph: *“America commissioned Musharraf with the task of taking revenge on the border tribes, especially the valiant and lofty Pashun tribes, in order to contain this popular support for jihad against its crusader campaign. So he began demolishing homes, making arrests, and killing innocent people. Musharraf, however, pretends to forget that these tribes, which have defended Islam throughout its history, will not bow to US”*

Our algorithm produces the following triplets for the example paragraph below:

Table 3. Extracted Triplets

Event	Subject	Verb	Object
	America	commission	Musharraf
	America	take	revenge
	Musharraf	demolish	homes
	Musharraf	make	arrests
	Musharraf	kill	innocent people
	Musharraf	pretend	<i>E1</i>
<i>E1</i>	Musharraf	forget	<i>E2</i>
<i>E2</i>	tribes	defend	Islam
<i>E2</i>	tribes	not bow	to US

4.3.1. Bottom-Up Event Tagging Approach. In the example above, consider the triplet <Musharraf, pretend, *E1*>. Here the object column of the verb *pretend* has an A1 argument including three other verbs (forget, defend and bow). That is, argument A1 is itself complex, comprising other triplets. So we tag

argument A1 with a nested event (*E1*), and recursively process A1 with our triplet extraction rules. We achieve this nested processing through a bottom-up algorithm that (i) detects simple verb occurrences (i.e. verbs with non-verb arguments) in the SRL parse tree, (ii) extracts triplets for those simple verb occurrences using the following **Triplet Matching Rules**, (iii) replaces simple verb clauses with an event identifier, thus turning all complex verb occurrences into simple verb occurrences with either non-verb or event arguments, and applies the following **Triplet Matching Rules**.

4.3.2. Triplet Matching Rules. We list four matching rules below to turn simple SRL columns into triplets:

- A0, V, A1: <SUBJECT, VERB, DIRECT OBJECT>
- A0, V, A2: <SUBJECT, VERB, PREPOSITION>, if direct object A1 not present in column.
- A0, V, A1, A2-AM-LOC: <SUBJECT, VERB, DIRECT OBJECT, location (PREPOSITION)>
- A1, V, A2: <DIRECT OBJ, VERB, PREPOSITION>

4.3.3. Triplet Extraction Accuracy. The triplet extraction accuracy is based on SRL accuracy. SRL has precision of 82.28%, recall of 76.78% and f-measure 79.44% [28].

4.3.4. Triplet Based Feature Extraction. For each verb (V) mentioned in a story (S), or non-story (NS) we stemmed and aggregated its arguments corresponding to its SUBJECTs, OBJECTs and PREPOSITIONs to generate following set-valued “semantic verb 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 extract its typed argument lists P.V.Subjects, P.V.Objects and P.V.Prepositions. Then, we match 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).

5. Generalized Verb Features

Generalization and reduction of features is an important step in classification process. Reduced feature representations not only reduce computing time but they

may also yield to better discriminatory behavior. Owing to the generic nature of the curse of dimensionality it has to be assumed that feature reduction techniques are likely to improve classification algorithm.

Our training data had 750 and 1,754 distinct verbs in stories and non-stories, yielding $750 * 3 = 2,250$ and, $1,754 * 3 = 5,262$ verb features for stories and non-stories respectively, and total of 7,512 features. VerbNet (VN) [29] is the largest on-line verb lexicon currently available for English. It is a hierarchical domain-independent, broad-coverage verb lexicon. VerbNet index has 5,879 total verbs represented, and these verbs are mapped into 270 total VerbNet main classes. For example, the verbs mingle, meld, blend, combine, decoct, add, connect all share the same meaning (i.e. to bring together or combine), and hence they map to verb class “mix” numbered 22.1. With the help of VerbNet and SRL argument types of the verbs, we mapped all occurrences of our verbs in stories and non-stories to one of these 270 VerbNet main classes. This mapping enabled us to reduce our verb features to $268 * 6 = 1,608$ verb features. The number 6 is used in the preceding equation since each verb class can lead to at most 6 features as V.Subject, V.Object and V.Preposition for its story and non-story occurrences. We started with 7,512 verb features, and after mapping these verb features to their verb category features we ended up with 1,608 features only.

In the generalization process, we faced a problem of verb sense disambiguation. There are some verbs which can be mapped to different senses, and each sense belongs to a different verb class. For example, the verb “add” can be used with the sense ‘mix’ (22.1) or ‘categorize’ (29.2) or ‘say’ (25.3). To solve this problem, we used argument types extracted using SRL for the ambiguous verbs. Then, we performed a look-up for each verb in the PropBank database to identify the matching verb sense with same type of arguments, and its verb class. PropBank [30] is a corpus that is annotated with verbal propositions, and their arguments - a “proposition bank”. In the look-up process, there is a chance that we may encounter more than one verb sense for the input verb matching the corresponding argument types. In this case, we picked the first matching verb sense listed in PropBank.

6. Experimental Evaluations

In this section, we evaluate the utility of standard keyword based features, statistical features based on shallow-parsing (such as density of POS tags and named entities), and a new set of semantic features to develop a story classifier. Feature extraction and

matching is implemented using JAVA and classification is performed using LIBSVM [22] in MATLAB.

Table 4. Classifier Performance for Stories

Feature Set	Precision	Recall	F-measure
POS	0.133	0.066	0.088
Keyword	0.821	0.205	0.329
Keyword + POS + NE	0.750	0.214	0.333
Triplet	0.798	0.515	0.626
Triplet + POS + NE	0.731	0.559	0.634

Table 5. Classifier Performance for Non-Stories

Feature Set	Precision	Recall	F-measure
POS	0.887	0.944	0.914
Keyword	0.903	0.994	0.946
Keyword + POS + NE	0.904	0.991	0.945
Triplet	0.886	0.998	0.938
Triplet + POS + NE	0.905	0.939	0.923

6.1. Effectiveness of Semantic Features

The baseline performance for a dummy classifier which would assign all instances to the majority class (non-story) would achieve 80.5% precision and 100% recall for the non-story category however, its precision and recall would be null for the stories. Hence, not useful at all for detecting stories.

Our proposed model makes use of triplets to incorporate both semantic and structural information available in stories and non-stories. In Table (4), we report the performance of SVM classification with various feature sets. SVM with POS, NE and generalized triplet based features **outperforms** other combinations of standard categories of features in terms of recall and F-measure. Table (4) shows 151.2% boost in recall and 90% boost in F-measure for keyword based (second row) vs triplet based (fourth row) features. After adding POS and NE features (Keyword + POS + NE based, third row) vs (Triplets + POS + NE, fifth row), we obtained 161.2% boost in recall and 90% boost in F-measure.

7. Conclusion

This paper proposes a model with semantic triplet based features for story classification. The effectiveness of the model is demonstrated against other traditional features used in the literature for text classification tasks. Future work includes more detailed evaluations, and also experiments with appropriate generalizations of nouns, adjectives and other types of keywords found in verb arguments.

Acknowledgment

This research was supported by an Office of Naval Research grant (N00014-09-1-0872) to the Center for Strategic Communication at Arizona State University.

References

- [1] J. Bruner and S. Weisser, "Autobiography and the construction of self," 1992.
- [2] C. Joseph, *The hero with a thousand faces*. Princeton University Press, 1949.
- [3] H. L. Halverson, J. R. Goodall and S. R. Corman, *Master Narratives of Islamist Extremism*. New York: Palgrave Macmillan, 2011.
- [4] R. Hoffman, U. Grasmann, R. Gueorguieva, D. Quinlan, D. Lane, and R. Mikkulainen, "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] J. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.
- [11] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proceedings of CoNLL-2003*, W. Daelemans and M. Osborne, Eds. Edmonton, Canada, 2003, pp. 142–147.
- [12] T. Poibeau and L. Kosseim, "Proper name extraction from non-journalistic texts," *Language and Computers*, vol. 37, no. 1, pp. 144–157, 2001.
- [13] J. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 363–370.
- [14] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Morristown, NJ, USA: Association for Computational Linguistics, 2009, pp. 147–155.
- [15] "Calais." [Online]. Available: <http://www.opencalais.com/>
- [16] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *In 6th Intl Semantic Web Conference, Busan, Korea*. Springer, 2007, pp. 11–15.
- [17] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for idf," *Journal of Documentation*, vol. 60, no. 5, pp. 503–520, 2004.
- [18] E. Brill, "A simple rule-based part of speech tagger," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 112–116.
- [19] T. Joachims, "A statistical learning model of text classification for support vector machines," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2001, pp. 128–136.
- [20] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.
- [21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [22] 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.
- [23] S. Keerthi and C. Lin, "Asymptotic behaviors of support vector machines with gaussian kernel," *Neural computation*, vol. 15, no. 7, pp. 1667–1689, 2003.
- [24] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenić, "Triplet extraction from sentences," *Proceedings of the 10th International Multiconference Information Society-IS*, pp. 8–12, 2007.
- [25] D. Hooge Jr, "Extraction and indexing of triplet-based knowledge using natural language processing," Ph.D. dissertation, University of Georgia, 2007.
- [26] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky, "Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task," *CoNLL 2011*, p. 28, 2011.
- [27] 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.
- [28] V. Punyakanok, D. Roth, and W. Yih, "The importance of syntactic parsing and inference in semantic role labeling," *Computational Linguistics*, vol. 34, no. 2, pp. 257–287, 2008.
- [29] K.ipper, A. Korhonen, N. Ryant, and M. Palmer, "A large-scale classification of english verbs," *Language Resources and Evaluation*, vol. 42, no. 1, pp. 21–40, 2008.
- [30] M. Palmer, D. Gildea, and P. Kingsbury, "The proposition bank: An annotated corpus of semantic roles," *Computational Linguistics*, vol. 31, no. 1, pp. 71–106, 2005.