A Hybrid Model and Memory Based Story Classifier

Betul Ceran, Ravi Karad, Steven Corman, Hasan Davulcu

Arizona State University
P.O. Box 87-8809
Tempe AZ 85287-8809
betul@asu.edu, rkarad@asu.edu, steve.corman@asu.edu, hdavulcu@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) based classifier.

Keywords: Story, Non-Story, Classification, Feature Extraction

1. Introduction

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. A story is defined as "a sequence of events leading to a resolution or projected resolution." 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 (Halverson and Corman, 2011) that interrelated stories that work together as a system are fundamental building blocks of (meta-) narrative analysis.

Computational models of stories have been studied for many different purposes. R.E. Hoffman et al. (2011) models stories using an artificial neural network. After the learning stage, they compare the story-recall performance of the neural network model with that of schizophrenic patients as well as normal controls. The most common form of classification applied on to the domain of 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 (Masand et al., 1992; Billsus and Pazzani, 1999).

Gordon investigated a similar problem to detect stories in conversational speech (Gordon and Ganesan, 2005) and weblogs(Gordon and Swanson, 2009). They use a confidence-weighted linear classifier with a variety of lexical features, and obtained the best performance with unigrams with precision = 66%, recall = 48%, F-score = 0.55. They applied this trained classifier (with 5002 blogs) to classify weblog posts in the ICWSM 2009 Spinn3r Dataset.

In this paper, we focus on discriminating between stories, and non-stories. The main contribution of this paper is the introduction of a new set of features based on 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 show that the combination of POS features, with semantic triplet-based features achieves highest accuracy with a Support Vector Machine (SVM) based classifier. We obtain precision of 0.706, recall of 0.559 and and F-measure of 0.634 which shows a 12% boost in precision and 5% boost in recall, an overall 10% boost in F-measure due to the utility of triplet based features.

2. System Architecture

2.1. 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. The texts come from groups including al-Qaeda, its affiliates, and groups known to sympathize with its cause. 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 the 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 Cohens Kappa = .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.

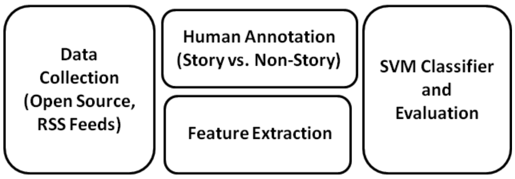
Data Collection (Open Source, RSS Feeds)
Human Annotation (Story vs. Non-Story)
Feature Extraction
SVM Classifier and Evaluation

Figure 1: System Architecture

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

This study only considers story vs. non-story codes. Our rationale for building a classifier using this data for training is as follows. A story typically contains several components. First, there must be an actor or actors. This can include politicians, mujahidin, and everyday people, etc. Second, the actors must be performing actions. This can include fighting, preparing for a battle, talking to others, etc. 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, victory, etc. Stories are differentiated from non-stories as following: 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.

2.3. Feature Extraction

In this paper we investigate the utility of standard keyword based features, statistical features based on shallow-parsing, and a new set of semantic features to develop a story classifier.

- **Keywords:** TF/IDF measure (Robertson, 2004) 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 (Brill, 1992) for each document is calculated with respect to numbers of tokens.

- **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.

2.4. Support Vector Machine (SVM) Classifier

SVM (Joachims, 2001) 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 (Boser et al., 1992) but later improved by the use of kernel functions to detect nonlinear patterns underlying the data (Cortes and Vapnik, 1995).There are various types of kernel functions available (Chang and Lin, 2011). 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 (Keerthi and Lin, 2003).

2.4.1. 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 (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.

3. 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 (Rusu et al., 2007; Jonnalagadda et al., 2009; Hooge Jr, 2007) 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).

3.1. 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 (Lee et al., 2011; Raghunathan et al., 2010) uses a heuristic approach to identify the noun phrases referred by the pronouns in a sentence. The heuristic is based on the num-

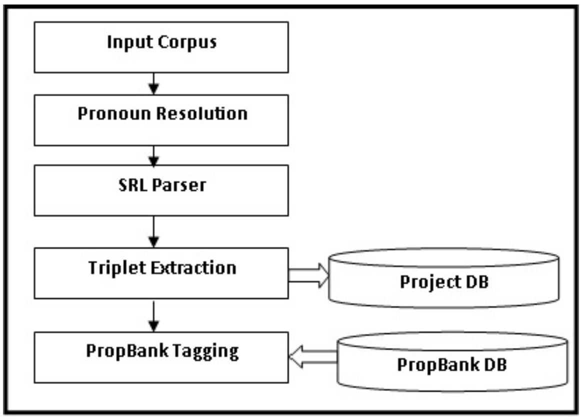

Figure 2: Triplet Extraction Pipeline

ber 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.

3.2. Semantic Role Labeler (SRL) Parser

SRL parser (Punyakanok et al., 2008) 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.

3.2.1. 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.

3.3. 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 Pashtun 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 above:

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

Table 1: Extracted Triplets

3.3.1. 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**.

3.3.2. Triplet Matching Rules

We list four matching rules below to turn simple SRL columns into triplets:

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

3.3.3. 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% (Punyakanok et al., 2008).

3.3.4. 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).

4. Generalized Verb Features

4.1. VerbNet(VN) Main Classes:

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) (Kipper et al., 2008) 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 previous 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 (Palmer et al., 2005) 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.

5. Experimental Evaluations

In this section, we evaluate the 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 (Chang and Lin, 2011) in MATLAB.

Feature Set	Precision	Recall	F-measure
POS	0.133	0.066	0.088
POS + Keywords	0.632	0.535	0.579
Triplets	0.548	0.321	0.405
POS + Triplets	**0.706**	**0.559**	**0.634**

Table 2: Classifier Performance for Stories

Feature Set	Precision	Recall	F-measure
POS	0.887	0.944	0.914
POS + Keywords	0.774	0.836	0.804
Triplets	**0.850**	0.936	**0.891**
POS + Triplets	0.805	**0.996**	0.891

Table 3: Classifier Performance for Non-Stories

5.1. 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 (2), we report the performance of SVM classification with various feature sets. SVM with POS and generalized triplet based features **outperforms** other combinations of standard categories of features in terms of precision and recall. If we compare the performance of POS features alongside keyword-based (second row) vs. triplet-based (fourth row) features, Table (2) shows 12% boost in precision and 5% boost in recall, resulting in 10% boost in F-measure for the story detection due to the utility of triplet based features.

6. Conclusion

This paper proposes a hybrid model with 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.

7. Acknowledgements

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.

8. References

D. Billsus and M.J. Pazzani. 1999. A hybrid user model for news story classification. *Lectures-International Centre for Mechanical Sciences*, pages 99–108.

B.E. Boser, I.M. Guyon, and V.N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

E. Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the workshop on Speech and Natural Language*, pages 112–116. Association for Computational Linguistics.

C.C. Chang and C.J. Lin. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

C. Cortes and V. Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Andrew S. Gordon and Kavita Ganesan. 2005. Automated story capture from conversational speech. In *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture*, page 145–152, Banff, Canada. ACM, ACM.

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

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

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

D.C. Hooge Jr. 2007. *Extraction and indexing of triplet-based knowledge using natural language processing*. Ph.D. thesis, University of Georgia.

T. Joachims. 2001. A statistical learning 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*, pages 128–136. ACM.

S. Jonnalagadda, L. Tari, J. Hakenberg, C. Baral, and G. Gonzalez. 2009. Towards effective sentence simplification for automatic processing of biomedical text. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 177–180. Association for Computational Linguistics.

S.S. Keerthi and C.J. Lin. 2003. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689.

K. Kipper, A. Korhonen, N. Ryant, and M. Palmer. 2008. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1):21–40.

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

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

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

V. Punyakanok, D. Roth, and W. Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

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

S. Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of Documentation*, 60(5):503–520.

D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenić. 2007. Triplet extraction from sentences. *Proceedings of the 10th International Multiconference Information Society-IS*, pages 8–12.