A Simpler and More Generalizable Story Detector using Verb and Character Features

Joshua D. Eisenberg and **Mark A. Finlayson**
11200 S.W. 8th Street, ECS Building, Miami, FL 33141
School of Computing and Information Sciences
Florida International University
Miami, Florida
{jeise003, markaf}@fiu.edu

Abstract

Story detection is the task of determining whether or not a unit of text contains a story. Prior approaches achieved a maximum performance of 0.66 F_1, and did not generalize well across different corpora. We present a new state-of-the-art detector that achieves a maximum performance of 0.75 F_1 (a 14% improvement), with significantly greater generalizability than previous work. In particular, our detector achieves performance above 0.70 F_1 across a variety of combinations of lexically different corpora for training and testing, as well as dramatic improvements (up to 4,000%) in performance when trained on a small, disfluent data set. The new detector uses two basic types of features–ones related to events, and ones related to characters–totaling 283 specific features overall; previous detectors used tens of thousands of features, and so this detector represents a significant simplification along with increased performance.

1 Motivation

Understanding stories is a long-held goal of both artificial intelligence and natural language processing. Stories can be used for many interesting natural language processing tasks, and much can be learned from them, including concrete facts about specific events, people, and things; commonsense knowledge about the world; and cultural knowledge about the societies in which we live. Applying NLP directly to the large and growing number of stories available electronically, however, has been limited by our inability to efficiently separate story from non-story text. For the most part, studies of stories *per se* has relied on manual curation of story data sets (Mostafazadeh et al., 2016), which is, naturally, time-consuming, expensive, and doesn't scale. These human-driven methods pay no attention to the large number of stories generated daily in news, entertainment, and social media.

The goal of this work is to build and evaluate a high performing story detector that is both simple in design and generalizable across lexically different story corpora. Our definition of story can be found in §1.2, and is based on definitions used in prior work on story detection. Previous approaches to story detection have relied on tens of thousands of features (Ceran et al., 2012; Gordon and Swanson, 2009), and have used complicated pre-processing pipelines (Ceran et al., 2012). Moreover these prior systems, while clearly important advances, did not, arguably, include features that captured the "essence" of stories. Furthermore, these prior efforts had poor generalizability, i.e. when trained on one corpus, the detectors perform poorly when tested on a different corpus. Building on this prior work, we begin to address these shortcomings, presenting a new detector that has many orders of magnitude fewer features than used previously, significantly improved cross corpus performance, and higher F_1 on all training and testing combinations.

1.1 Task

Our goal is to design a system that can automatically decide whether or not a paragraph of text contains a story. We say a paragraph contains a story if any portion of it expresses a significant part of a story, including the characters and events involved in major plot points. Corpora used in prior work included Islamic Extremist texts (Ceran et al., 2012), and personal weblog posts (Gordon and Swanson, 2009), which were both annotated at

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2698–2705
Copenhagen, Denmark, September 7–11, 2017.

this level of granularity. In this paper we test combinations of new features on both of these corpora. Once we determined the best-performing feature set, we ran experiments using those features to evaluate its generalizability across corpora.

1.2 What is a Story?

Author E.M. Forster said "A story is a narrative of events arranged in their time sequence" (Forster, 2010). A more precise definition, of our own coinage, is that a narrative is a discourse presenting a coherent sequence of events which are causally related and purposely related, concern specific characters and times, and overall displays a level of organization beyond the commonsense coherence of the events themselves. In sum, a story is a series of events effected by animate actors. This reflects a general consensus among narratologists that there are at least two key elements to stories, namely, the plot (*fabula*) and the characters (*dramatis personae*) who move the plot forward (Abbott, 2008). While a story is more than just a plot carried out by characters–indeed, critical to 'storyness' is the connective tissue between these elements that can transport an audience to a different time and place–here we focus on these two core elements to effect better story detection.

1.3 Outline of the Paper

We begin by discussing prior work on story detection (§2). Then we introduce our new detector (§3), which relies on simple verb (§3.1) and character (§3.2) features. We test our detector on two corpora (§3.3)—one of blog posts and one of Islamist Extremist texts—using an SVM model to classify each paragraph as to whether or not it contains a story (§3.4). We conduct an array of experiments evaluating different combinations and variants of our features (§4). We also detail our use of undersampling for the majority class (§4.1), as well as our cross validation procedure (§4.2). We present both the results of the single corpus experiments (§4.3) and the cross-corpus and generalizability experiments (§4.4). We conclude with a list of contributions and discussion of future directions (§5).

2 Related Work

There have been three major contributions to the study of automatic story detection. In 2009, Gordan and Swanson developed a bag-of-words-based detector using blog data (Gordon and Swanson, 2009). They annotated a subset of paragraph-sized posts in the Spinn3r Blog corpus for the presence of stories, and used this data to train a confidence weighted linear classifier using all unigrams, bigrams, and trigrams from the data. Their best F_1 was 0.55. This was an important first step in story detection, and the annotated corpus of blog stories is an invaluable resource.

In 2012, Corman *et al.* developed a semantic-triplet-based detector using Islamist Extremist texts (Ceran et al., 2012). They annotated paragraphs of the CSC Islamic Extremist corpus for the presence of stories, and used this data to train an SVM with a variety of features including the top 20,000 *tf-idf* tokens, use of stative verbs, and agent-verb-patient triplets ("semantic triplets," discussed in more detail below in §3.1). Their best performing detector in that study achieved 0.63 F_1. The intent of the semantic triplet features was to encode the plot and the characters. These features were intended to capture the action of stories, but the specifics of the implementation was problematic: each unique agent-verb-patient triplet has its own element in the feature vector, and so this detector was sensitive primarily to the words that appeared in stories, not generalized actions or events.

Although Corman's detector has a higher F_1 than Gordon's, it was not clear which one was actually better; they were tested on different corpora. We compared the two detectors by reimplementing both, confirmed the correctness of the reimplementations, and running experiments where each detector was trained and tested on the corpora (Eisenberg et al., 2016). After these experiments, we showed that Corman's detector had better performance on the majority of experiments. Some of the results of these experiments are shown in Table 2. We also slightly improved the performance of Corman's detector to 0.66 F_1. In addition we reported results investigating the generalizability of the detectors; these results showed that neither the Gordon nor the Corman detectors generalized across corpora. We ascribed this problem to the fact that the features of each detector were closely tied to the literal words used, and did not attempt to generalize beyond those specific lexical items.

In terms of domain independence, we surveyed other discourse related tasks to see how generalization across domains has been achieved. For

example, Braud *et al.* achieved domain independence in the identification of implicit relations between discourse units by training their system on both natural and synthetic data, weighting the influence of the two types (Braud and Denis, 2014). Jansen *et al.*, as another example, demonstrated domain independence on the task of non-factoid question answering by using both shallow and deep discourse structure, along with lexical features, to train their classifiers (Jansen et al., 2014). Thus, domain independence is certainly possible for discourse related tasks, but there does not yet seem to be a one-size-fits-all solution.

3 Developing the Detector

In contrast to focusing on specific lexical items, our implementation focuses on features which we believe capture more precisely the essence of stories, namely, features focusing on (a) events involving characters, and (b) the characters themselves.

3.1 Verb Features

Verbs are often used to express events. We use this fact to approximate event detection in a computationally efficient but still relatively accurate manner. The first part of each feature vector for a paragraph comprises 278 dimensions, where each element of this portion of the vector represents one of the 278 verb classes in VerbNet (Schuler, 2005). The value of each element depends on whether a verb from the associated verb class is used in the paragraph. Each element of the vector can have three values: the first value represents when a verb from the element's corresponding verb class is used in the paragraph and also involves a character as an argument of the verb. The second value represents when a verb from the verb class is used, but there are no characters involved. The third value represents the situation where no verbs from the verb class are used in the paragraph.

For clarity, we list the general steps of the verb feature extraction pipeline:

1. Split each paragraph into tokens, assign part of speech tags, and split the text into sentences, all using Stanford CoreNLP (Manning et al., 2014).
2. Parse each sentence with OpenNLP (Apache Foundation, 2017).
3. Label each predicate with its semantic roles using the SRL from the Story Workbench (Finlayson, 2008, 2011).
4. Disambiguate the Wordnet sense (Fellbaum, 1998) for each open-class word using the *It Makes Sense* WSD system (Zhong and Ng, 2010), using the Java WordNet Interface (JWI) to load and interact with WordNet (Finlayson, 2014).
5. Assign one of 278 VerbNet verb classes to each predicate, based on the assigned Wordnet sense, and using the jVerbnet library to interact with VerbNet. (Finlayson, 2012).
6. Determine whether the arguments of each predicate contains characters by using the Stanford Named Entity Recognizer (Finkel et al., 2005) and a gendered pronoun list.

We considered an argument to involve a character if it contained either (1) a gendered pronoun or (2) a named entity of type *Person* or *Organization*. We treated organizations as characters because they often fulfill that role in stories: for example, in the Extremist stories, organizations or groups like the *Islamic Emirate*, *Hamas*, or *the Jews* are agents or patients of important plot events. The verb features were encoded as a vector with length 278, each entry representing a different VerbNet verb class with three possible values: the verb class does not appear in the paragraph; the verb class appears but does not involve characters; or the verb class appears and a character is either an agent, patient, or both.

The verb features represent the types of events that occur in a paragraph, and whether or not characters are involved in those events. This is a generalized version of the semantic triplets that Corman *et al.* used for their story detector (Ceran et al., 2012), where they paired verbs with the specific tokens in the agent and patient arguments. The disadvantage of Corman's approach was that it led to phrases with similar meaning being mapped to different features: for example, the sentences "Bob played a solo" and "Mike improvised a melody" are mapped to different features by the semantic triplet based detector, even though the meaning of the sentences are almost the same: a character is performing music. On the other hand, in our approach, when we extract verb feature vectors from these sentences, both result in the same feature value, because the verbs *played* and *improvised* belong to the *performance* VerbNet class, and both verbs have a character in one of their arguments. This allow a generalized encoding of the types of

action that occurs in a text.

3.2 Character Features

Our second focus is on character coreference chains. Characters, as discussed previously, are a key element of stories. A character must be present to drive the action of the story forward. We hypothesize that stories will contain longer coreference chains than non-stories. To encode this as a feature, we calculated the normalized length of the five longest coreference chains, and used those numbers as the character features. We computed these values as follows:

1. Extract coreference chains from each paragraph using Stanford CoreNLP coreference facility (Clark and Manning, 2016).
2. Filter out coreference chains that do not contain a character reference as defined in the Verb section above (a named entity of type *Person* or *Organization*, or a gendered pronoun).
3. Sort the chains within each paragraph with respect to the number of references in the chain.
4. Normalize the chain lengths by dividing the number of referring expression in each chain by the number of sentences in the paragraph.

These normalized chain lengths were used to construct a five-element feature vector for use by the SVM. We experimented with different numbers of longest chains, anywhere from the single longest to the ten longest chains. Testing on a development set of 200 Extremist paragraphs revealed using the five longest chains produced the best result.

3.3 Corpora

As noted, we used two corpora that were annotated by other researchers for the presence of stories at the paragraph level. The CSC Islamic Extremist Corpus comprises 24,009 paragraphs (Ceran et al., 2012), of which 3,300 were labeled as containing a story. These texts recount Afghani and Jihadi activities in the mid-2000's in a variety of location around the world. This corpus was originally used to train and test Corman's semantic-triplet-based story detector. The web blog texts come from the ICWSM 2009 Spinn3r Dataset (Burton et al., 2009). The full data set contains 44 million texts in many languages. Gordon and Swanson (2009) annotated a sample of 4,143 English texts from the full data set, 201 of which were identified as containing stories. This corpus was originally used to train and test Gordon's bag-of-words-based detector. Most of the texts in the blog corpus are no more than 250 characters, roughly a paragraph. The distribution of texts can be seen in Table 1.

Corpus	Story	Non-Story
Extremist	3,300	20,709
Blog	201	3,942

Table 1: Distribution of story paragraphs across the Extremist and blog corpora.

3.4 SVM Machine Learning

We used the Java implementation of LibSVM (Chang and Lin, 2011) to train an SVM classifier with our features. The hyper-parameters for the linear kernel were $\gamma = 0.5$, $\nu = 0.5$, and $c = 20$.

4 Experiments & Results

The results of our new experiments are shown in Table 3. We report precision, recall, and F_1 relative to the story and non-story classes. We performed experiments on three feature sets: the verb features alone (indicated by **Verb** in the table), character features alone (indicated by **Char**), and all features together (**Verb+Char**). We conducted experiments ranging over three corpora: the Extremist corpus (**Ext**), the blog corpus (**Web**), and the union of the two (**Comb**). These results may be compared with the previously best performing detector, namely, Corman's semantic triplet based detector (Ceran et al., 2012), as tested by us in prior work (Eisenberg et al., 2016), and shown in Table 2.

Training	Testing	Prec.	Recall	F_1
Ext	**Ext**	0.77	0.57	**0.66**
Ext	**Web**	0.23	0.37	0.28
Ext	**Comb**	0.43	0.41	0.32
Web	**Web**	0.66	0.31	0.43
Web	**Ext**	0.59	0.003	0.01
Web	**Comb**	0.59	0.01	0.01
Comb	**Ext**	0.62	0.51	0.43
Comb	**Web**	0.36	0.49	0.30
Comb	**Comb**	0.64	0.47	0.46

Table 2: Results for the Corman semantic triplet based detector as reported in (Eisenberg et al., 2016). These results are with respect to the story class.

			Not Story			Story		
Features	**Training**	**Testing**	Prec.	Recall	F_1	Prec.	Recall	F_1
Verb	**Ext**	**Ext**	0.73	0.81	0.77	0.78	0.70	0.74
Verb	**Web**	**Web**	0.69	0.75	0.72	0.73	0.66	0.69
Char	**Ext**	**Ext**	0.30	0.27	0.21	0.52	0.74	0.55
Char	**Web**	**Web**	0.67	0.68	0.67	0.67	0.65	0.65
Verb+Char	**Ext**	**Ext**	0.73	0.81	0.77	0.79	0.70	**0.74**
Verb+Char	**Ext**	**Web**	0.68	0.80	0.73	0.75	0.63	0.69
Verb+Char	**Ext**	**Comb**	0.70	0.77	0.73	0.75	0.67	0.71
Verb+Char	**Web**	**Web**	0.71	0.76	0.72	0.74	0.68	0.70
Verb+Char	**Web**	**Ext**	0.50	0.82	0.62	0.50	0.18	0.27
Verb+Char	**Web**	**Comb**	0.53	0.79	0.64	0.60	0.40	0.41
Verb+Char	**Comb**	**Ext**	0.74	0.81	0.77	0.79	0.71	**0.75**
Verb+Char	**Comb**	**Web**	0.68	0.74	0.70	0.72	0.64	0.67
Verb+Char	**Comb**	**Comb**	0.72	0.81	0.76	0.79	0.68	0.73

Table 3: Results of the new detectors as trained and tested on the Extremist (Ext), weblog (Web), or combined (Comb) corpora. The feature sets tested include the 278 verb class features (Verb), the normalized length of the five longest coreference chains (Char), and the combination of these two feature sets (Verb+Char). Undersampling is utilized in each of these experiments.

4.1 Undersampling

In each of the new experiments, we undersampled the non-story class before training (Japkowicz, 2000). Undersampling is a technique used to help supervised machine learning classifiers learn more about a class that has a significantly smaller number of examples relative to an alternative. In our case, non-story labels outnumbered story labels by a factor of 7 overall. Extremist story paragraphs are only 15.9% of the total annotated paragraphs in that set, and in the blog corpus stories were only 4.9% of the paragraphs. To prevent the detector from being over trained on non-story paragraphs, we thus reduced the size of the non-story training data to that of the story data, by randomly selecting a number of non-story texts equal to the number of story texts for training and testing.

4.2 Cross Validation

We used three versions of cross validation for the new experiments, one for each experimental condition: training and testing on a single corpus; training on a single corpus and testing on the combined corpus; or training on the combined corpus and testing on a single corpus. These procedures are the same as in our previous work (Eisenberg et al., 2016). We performed undersampling before cross validation, so when we are explaining how to divide up the story and non-story texts into cross validation folds, this refers to the full set of story texts and the set of non-story texts that was randomly selected to equal the number of story texts. For all experiments with cross validation, we use ten folds.

Train and Test on a Single Corpus: If the training and testing corpus is the same, divide up the stories into ten subsets of equal size, and the undersampled non-stories into ten subsets of equal size. For each fold of cross validation a different story set and non-story set (of the same index) are used as the testing set and the remaining nine are used for training.

Train on Combined, Test on Single: If the training is done on the combined corpus, and the test corpus is either the weblog or Extremist corpus, which we will refer to as the single test corpus, first divide the stories from the single test corpus into ten equal sized sets, and then divide up that corpus's non-stories into ten equal sets. For each fold of cross validation a different story set and non-story set (of the same index) from the single test corpus are used as the testing set and the remaining nine are used for training. The texts from the other corpus (the corpus that is not the single test corpus), are undersampled and added to all ten folds of training.

Train on Single, Test on Combined: If training is done on a single corpus, and the test corpus is the combined corpus, first divide the stories from the single training corpus into ten equal sized sets, and the undersampled non-stories from the single training corpus into ten equal sized sets. For each fold of cross validation a different story set and non-story set (of the same index) from the single training corpus are used as the testing set and the remaining nine are used for training. The texts from the other corpus (the corpus that is not

the single training corpus), are undersampled and added to all ten folds of testing.

4.3 Single Corpus Experiments

For every experiment that used only a single corpus, the best feature set included both the verb and character features, achieving up to 0.74 F_1 when trained and tested on the Extremist corpus. This represents a new state-of-the-art, about 12.6% greater than the performance of Corman's detector when trained and tested on the same corpus (0.66 F_1).

When the detector uses only verb features it achieves an F_1 of 0.74 on the Extremist corpus, only 0.002 lower than the detector using all the features. Interestingly, the detector achieves 0.55 F_1 using only the five character features, which is respectful given such a small feature set. To put this in perspective, the Corman detector (Ceran et al., 2012) uses more than 20,000 features, and achieves an F_1 of 0.66. Thus we were able to achieve 83% of the performance of the Corman detector with 4,000 times fewer features.

When training and testing on the blog corpus, the detector using all the features achieved 0.70 F_1, a 74% increase from the Corman detector's 0.425 F_1. This is the best performing model on the blog corpus, from any experiment to date. The detector using only verb features achieves 0.74 F_1, which is only slightly worse than when both sets of features are used. When we trained using only the character features, the system achieves 0.65 F_1, which is still 54% higher than when the Corman detector is trained and tested on the blog corpus.

In the single corpus experiments, the detectors that we trained and tested on the Extremist paragraphs have higher performance than those trained on the web blogs, except for when we use only the five character features. A possible reason for this is the Stanford NER may not be recognizing the correct named entities in the Extremist texts, which contain many non-Western names, e.g., *Mujahidin*, *Okba ibn Nafi*, or *Wahid*. However, when we include the verb features, the detectors trained on the Extremist texts achieve better performance. We believe this is partially due to the greater number of stories in the Extremist corpus, and their increased grammatical fluency. The Extremist corpus is actually well written compared to the blog corpus, the latter of which contains numerous fragmentary and disjointed posts.

4.4 Cross Corpus Experiments

We show the generalizability of our best-performing detector (that including both verb and character features) by training it on one corpus and testing it on another.

When we trained the detector on the Extremist texts and tested on the blog texts, it scores a 0.68 F_1. This is 142% improvement over Corman's detector in the same setup (0.28 F_1), and is a higher F_1 than the previous state-of-the-art on any single corpus test. When we trained the detector on the Extremist corpus and tested on the combined corpus, it achieved 0.71 F_1, which is an 121% increase from Corman's detector in the equivalent setup.

For the detector trained on the blog corpus and tested on the Extremist corpus, the detector that uses both verbs and character features achieves an 0.27 F_1, which is a 2,600% increase over the Corman detector's 0.01 F_1 in this same setup. While 0.27 F_1 can by no means be called good performance, it is significantly better than the Corman detector's performance on this task, and so demonstrates significantly better generalizability. As seen in our experiments, detectors trained on only the blog corpus do not perform as well as detectors trained on the Extremist corpus. We suspect that this is partially due to the disfluent nature of the blog corpus, which includes many fragmentary sentences, grammatical errors, and slang, all of which are difficult for the NLP pipeline to handle.

Note that we performed no cross validation in the above experiments where we trained the detector on the Extremist corpus and tested on the blog corpus, or vice versa, because in these cases the training and testing sets have no intersection.

The cross corpus experiment with the largest percent increase is for the verb and character detector trained on the blog corpus and tested on the combined corpus. The new detector's F_1 is 0.41, a 4,000% increase from the Corman detector's 0.01 F_1 on this task. Although a 0.41 F_1 is also not good, this is a massive improvement over previous performance. This is further evidence that our verb and character feature based detector is significantly more generalizable than Corman's approach.

The remaining five cross corpus experiments involved the combined corpus. In this case, our detector out-performed Corman's detector. Of

special note is the detector trained on the combined corpus and tested on the Extremist corpus. It achieved 0.75 F_1, which is 0.01 points of F_1 higher than our best single corpus detector, which was trained and tested on the Extremist corpus. This isn't a substantial increase in performance, but it suggests that information gleaned from the blog corpus does potentially–albeit marginally–help classification of the Extremist texts.

5 Conclusion

We have introduced a new story detection approach which uses simple verb and character features. This new detector outperforms the prior state-of-the-art in all tasks, sometimes by orders of magnitude. Further, we showed that our detector generalizes significantly better across lexically different corpora. We propose that this increase in performance and generalizability is due to the more general nature of our features, especially those related to verb classes. This approach has additional advantages, for example, the feature vector is fixed in size and does not grow in an unbounded fashion as new texts (with new verbs, agents, and patents) are added to the training data.

In future work we plan to develop richer character-based features. The current approach uses only normalized lengths of the five longest coreference chains, which leaves out important information about characters that could be useful to story detection. Indeed, our experiments showed that these character features only add a small amount of information above and beyond the verb features. However, when used alone, the character features still yield reasonable performance, which suggests that more meaningful character-based features could lead to story detectors with even better performance.

Acknowledgments This work was partially supported by National Institutes of Health (NIH) grant number 5R01GM105033-02. We thank W. Victor H. Yarlott for building a wrapper class to the *It Makes Sense* WSD. We also thank Reid Swanson especially, as well as Andrew Gordon, for working to retrieve, format, and providing their original annotations of the Weblog corpus. Thanks to Steve Corman for facilitating the transmission of the Extremist story data, which is covered by U.S. government FOUO rules.

References

H. Porter Abbott. 2008. *The Cambridge Introduction to Narrative*. Cambridge University Press, Cambridge, England.

Apache Foundation. 2017. OpenNLP v1.6.0, https://opennlp.apache.org, Last accessed on July 20, 2017.

Chloé Braud and Pascal Denis. 2014. Combining Natural and Artificial Examples to Improve Implicit Discourse Relation Identification. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1694–1705, Dublin, Ireland.

Kevin Burton, Akshay Java, and Ian Soboroff. 2009. The ICWSM 2009 Spinn3r Dataset. In *Proceedings of the 3rd Annual Conference on Weblogs and Social Media (ICWSM 2009)*, San Jose, CA.

Betul Ceran, Ravi Karad, Steven Corman, and Hasan Davulcu. 2012. A Hybrid Model and Memory Based Story Classifier. In *Proceedings of the 3rd International Workshop on Computational Models of Narrative (CMN'12)*, pages 60–64, Istanbul, Turkey.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27.

Kevin Clark and Christopher D. Manning. 2016. Deep Reinforcement Learning for Mention-Ranking Coreference Models. In *Proceedings of the 21st Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 2256–2262, Austin, Texas.

Joshua D Eisenberg, W Victor H Yarlott, and Mark A Finlayson. 2016. Comparing extant story classifiers: Results & new directions. In *Proceedings of the 7th International Workshop on Computational Models of Narrative (CMN'16)*, Paper 6, Krakow, Poland.

Christine Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, pages 363–370, Ann Arbor, MI.

Mark A. Finlayson. 2008. Collecting Semantics in the Wild: The Story Workbench. In *Proceedings of the AAAI Fall Symposium on Naturally Inspired Artificial Intelligence*, pages 46–53, Arlington, VA.

Mark A. Finlayson. 2011. The Story Workbench: An Extensible Semi-Automatic Text Annotation Tool. In *Proceedings of the 4th Workshop on Intelligent Narrative Technologies (INT4)*, pages 21–24, Stanford, CA.

Mark A. Finlayson. 2012. JVerbnet, v1.2.0, http://projects.csail.mit.edu/jverbnet, Last accessed on July 17, 2017.

Mark A. Finlayson. 2014. Java Libraries for Accessing the Princeton Wordnet: Comparison and Evaluation. In *Proceedings of the 7th International Global Wordnet Conference (GWC 2014)*, pages 78–85, Tartu, Estonia.

Edward M. Forster. 2010. *Aspects of the Novel*. RosettaBooks, New York City, New York.

Andrew Gordon and Reid Swanson. 2009. Identifying Personal Stories in Millions of Weblog Entries. In *Proceedings of the 3rd International Conference on Weblogs and Social Media (ICWSM 2009), Data Challenge Workshop*, pages 16–23, San Jose, CA.

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse Complements Lexical Semantics for Non-factoid Answer Reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), Long Papers*, pages 977–986, Baltimore, MD.

Nathalie Japkowicz. 2000. Learning from Imbalanced Data Sets: a Comparison of Various Strategies. In *AAAI Workshop on Learning from Imbalanced Data Sets*, pages 10–15, Austin, TX.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), System Demonstrations*, pages 55–60, Baltimore, MD.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Karin Kipper Schuler. 2005. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

Zhi Zhong and Hwee Tou Ng. 2010. It Makes Sense: A Wide-Coverage Word Sense Disambiguation System for Free Text. In *Proceedings of the 48th Annual Meeting of the Assocation for Computational Linguistics (ACL 2010), System Demonstrations*, pages 78–83, Uppsala, Sweden.