WriteRec

Ariella Levine, Clay Riley, Patricia Whitlock
Brandeis University, Brandeis University, Brandeis University
Cambridge, MA., Roslindale, MA., Waltham, MA.
arilevine@brandeis.edu, riley@brandeis.edu, plw5406@brandeis.edu

Abstract

An annotation project was designed and carried out to create a corpus of attributed directed non-acyclic graphs representing characters (nodes) and the relationships among them (edges) in plot summaries of books with the goal of making recommendations to readers based on similarities with the social networks of books they already prefer. Cycles of specification, guideline generation, corpus document curation, annotation, receipt of feedback from annotators, and reworking were completed in order to develop a corpus with a design that could be consistently recreated. A single genre, heroic fantasy, was used to promote consistency and allow for detailed specification tailored to common formats of the genre's plot structures. Annotators were students taking a course in annotation specification and design. Once a gold standard corpus of plot summary social networks was established, k-means clustering was used to establish groupings of books in which characters interact similarly. A baseline was also created using clustering over binary unigram features.

Keywords: book recommendations, plot summaries, character descriptions

1. Task Goals

There were three goals in this project. First, to acquire plot summaries of uniform size and format for one genre. Second, to identify entities as well as the events, properties, and relationships that affect the entities and show the interactions between the entities. The third goal was to recommend books based on similarities in plots, characters, and character development.

2. Annotation Specifications

The key parts of the social network being generated during annotation are the characters and other entities and the relationships between them. Entities are nodes in the graph; relations make up the edges connecting them. Words identifying the presence of an edge were also to be collected. For this annotation process the annotator is responsible for identifying entities, triggers, and identifying relations between entities via triggers.

3. Annotation Instructions

3.1. Entity Identification and Tagging

The annotator should tag each unique character and object. As well as, mark attributes that pertain to each character and object such as subtype, number etc. For full description of characters and objects refer to Entity Identification, Annotation Guidelines.

3.2. Trigger Identification and Tagging

The annotator should tag verbs, descriptions, and other events that represent the interactions or relationships between characters. Annotators should also tag verbs, and other events that represent internal or external struggles, victories, and failures between characters and their environment. Lastly, for each tag the annotator should mark the required attributes. For full description see section Relation Trigger Identification, Annotation Guidelines.

3.3. Identifying Relations between Entities and Tagging

The annotator should show relations between characters by creating a relational link tag. The way to do this is by creating a link tag with no span selected. Then selecting the trigger that defines the relation. Next, selecting the entity from which the relation is caused or directed and label it "From". Lastly, select the entity to which the relation is directed or given and label it "To". Finally, for each relation the annotator should mark the required attributes for each tag. For full description see section Relational Linking, Annotation Guidelines.

4. Characteristics of the Data Set

We chose to work on novels from the genre of 'heroic fantasy' and to use the information from goodreads.com. We organized a collection of 150 book summaries from goodreads.com. Weeks 1, 2, and 3 each had 50 book summary documents to annotate. The summaries had nonstandard length and nonstandard content. Some had more relevant information to the task and some had less relevant information to the task.

GoodReads allows book summaries to be either from the publisher or written by GoodReads users. Publisher information is already non-standard, for publisher publisher on the back of the book or inside the front cover whatever they think will sell. GoodReads also allows for users to write plot summaries with no apparent guidelines. Which is to say, summaries were all over the place. Some included irrelevant publishing information which we asked annotated to ignore. Some included no information whatsoever about characters and relationships. Some plot summaries went into more information about relationships and some went into less information about relationships. The final size of the gold standard corpus was 136 documents.

5. Difficulties Collecting the Data Set

5.1. Data Source

Collecting summaries from GoodReads for the "Heroic Fantasy" genre proved to be a challenge in that the way the website was designed made it hard to find books in the correct genre. We had thought that the website had a master list of all titles in the genre, but the list turned out to reflect only a small subset of the genre. Additionally, many of the titles in the genre were from the same series and we decided to not include novels in the same series in our data set, as their plot summaries would be exceedingly similar to each other.

5.2. Span Issues

There were issues reading the files in MAE due to problems with span. Text files that had new line characters resulted in disrupted alignment of the spans. This caused difficulties in abjudication, as it was difficult to see whta was going on.

5.3. Implicating Updated DTD

Difficulties arose when trying to update old xml files to match the most recent updated version of the DTD. This should have been easier than it turned out to be. MAE was not very helpful as it often crashed when the xml file did not match the DTD perfectly and it was difficult to always spot the issue and deal with the issue properly.

5.4. Low Annotator Agreement

Due to having a small number of annotators, the agreement between annotators was low.

5.5. Abjudication

IAA on relations (edges) cannot be automatically calculated and requires manually aligning and verifying each edge for each copy of each document.
The annotation and adjudication tool used forces distinction between different types of entities, decreasing IAA metrics artificially.

5.6. Evaluating Results

As this was a clustering and not a classification machine learning task, intrinsic evaluation proved to be a difficult problem. Holding out test data was not an option, as testing in this way would not show whether or not the clusters were identifying useful recommendations. Proper intrinsic evaluation would require a large number of document raters rating their interest in every document, and using these aggregated statistics to generate clusters based on ratings. However, the time and resources did not exist for this to come to fruition. Instead, evaluation took the form of spot checking the clusters by taking reader ratings for cluster exemplars and comparing the ratings.

5.7. Possible Improvements

Find a better data source that contains a larger amount of the "Heroic Fantasy" genre. Updating Updating old xml files could be done with more time allocated to this issue. Low annotator agreement can possibly be fixed with having a larger group of annotators. Span and abjudication issues can possibly be fixed with future version sof MAE. A new iteration of this project would make all entity types a single tag, "entity/character", and all current tag types (e.g. "Protagonist") attributes thereof.

6. Annotation Quality

6.1. Metrics Used to Measure Quality and Justification

6.1.1. Krippendorff's Alpha

The researchers of this experiment decided on Krippendorff's alpha because it is a good option for when there is sparse agreement between annotators which ultimately was a factor in this experiment. Another factor in deciding to use Krippendorff's alpha was that MAE partially implements Krippendorff's alpha; therefore, it was easier for the researchers of this experiment to measure annotator agreement.

6.1.2. Fleiss' Kappa

In addition to implementing Krippendorff's alpha the researchers of this experiment chose to implement Fleiss's kappa as well for the following reasons. Similar to Krippendorff's alpha MAE partially implements Fleiss's kappa; therefore, it was easier for the researchers to calculate the agreement between multiple annotators.

6.2. Interpretation of Numerics and Error Analysis

All IAA values represent the specified metric as calculated by MAE v2.1.4 on the set of only and all documents that became part of the gold standard. Three annotators' copies exist for all but one document, which has two annotators' copies.
The cross-tag agreement for segmentation (span selection) for all tags besides triggers: ==

```
<Cross-tag> Alpha-U (Krippendorf's)
[Antagonist, Object, Other, Protagonist,
Unkown] ==0.4302 (<Cross-tag> Alpha-U
(Krippendorf's)  [Antagonist, Object,
Other, Protagonist, Unkown])
cross-tag_alpha_u
```

The cross-tag agreement for segmentation (span selection) for triggers: ==

```
<Cross-tag> Alpha-U (Krippendorf's)  [Trigger]
== 0.3546 (<Cross-tag> Alpha-U (Krippendorf's)
[Trigger]) cross-tag_alpha_u
```

The agreement for each tag's segmentation (span selection): ==

```
<Tag-level> Alpha-U (Krippendorf's) ==
 0.4624 (<Tag-level> Alpha-U (Krippendorf's))
Antagonist::-
 0.4222 (<Tag-level> Alpha-U (Krippendorf's))
Object::-
 0.3857 (<Tag-level> Alpha-U (Krippendorf's))
Other::-
 0.4932 (<Tag-level> Alpha-U (Krippendorf's))
Protagonist::-
 0.3546 (<Tag-level> Alpha-U
```

```
(Krippendorf's)) Trigger::-
-0.0036 (<Tag-level> Alpha-U
(Krippendorf's)) Unkown::-
```

The agreement for each tag attribute's segmentation–how well the annotators agreed on an attribute for a given segment: ==

```
<Tag-level> Alpha-U (Krippendorf's) ==
 0.4293 (<Tag-level> Alpha-U
(Krippendorf's)) Antagonist::number
 0.3019 (<Tag-level> Alpha-U
(Krippendorf's)) Antagonist::subtype
 0.3817 (<Tag-level> Alpha-U
(Krippendorf's)) Object::number
 0.4055 (<Tag-level> Alpha-U
(Krippendorf's)) Other::number
 0.3068 (<Tag-level> Alpha-U
(Krippendorf's)) Trigger::subtype
 0.4196 (<Tag-level> Alpha-U
(Krippendorf's)) Protagonist::number
 0.2529 (<Tag-level> Alpha-U
(Krippendorf's)) Protagonist::subtype
```

The agreement for labels for each tag type and attribute: ==

```
<Tag-level> Multi-Pi (Fleiss' Kappa) ==
-0.1918 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Antagonist::-
 0.5782 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Antagonist::number
 0.3766 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Antagonist::subtype
-0.2621 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Object::-
 0.5305 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Object::number
-0.2736 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Other::-
 0.6342 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Other::number
-0.0324 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Protagonist::-
 0.9259 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Protagonist::number
 0.4130 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Protagonist::subtype
-0.2600 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Trigger::-
 0.6613 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Trigger::subtype
-0.5000 (<Tag-level> Multi-Pi
(Fleiss' Kappa)) Unkown::-
```

The cross-tag agreement for labels of character entities: ==

```
<Cross-tag> Multi-Pi (Fleiss' Kappa)
[Antagonist, Object, Other, Protagonist,
Unkown] ==
0.1358 (<Cross-tag> Multi-Pi (Fleiss'
Kappa)  [Antagonist, Object, Other,
Protagonist, Unkown]) cross-tag_multi_pi
```

Inter-annotator agreement is particularly low for various reasons. This likely stems from the low number of annotators as well as issues understanding the guidelines due to insufficient cycles of reworking. The specification also does not account for the software used to generate the annotations. As all character types are given their own tag label, they show extremely poor segmentation agreement; even when every annotator identified and selected a character correctly, if they each chose a different highest-level label, agreement is not easily shown by automated means.

7. Machine Learning Experiment Design

7.1. Baseline System and Baseline Features

Binarized "bag of words" was used with k-means clustering to create a baseline. By binarizing the vector dimensions, there are effects only for the presence or absence of a word. In this way common words appearing in every text do not increase the distance between smaller and larger documents. Another technique to achieve slightly more nuanced results would be to use tf-idf to weight non-binarized vector values. bow_clusters = [3, 2, 0, 4, 4, 0, 4, 0, 0, 3, 2, 3, 3, 0, 4, 4, 4, 3, 2, 0, 4, 3, 4, 3, 2, 3, 0, 2, 0, 2, 0, 0, 4, 0, 0, 2, 4, 4, 2, 0, 4, 4, 4, 4, 2, 2, 0, 0, 4, 4, 3, 0, 4, 2, 3, 1, 0, 0, 0, 3, 4, 0, 4, 4, 0, 2, 2, 4, 0, 0, 0, 1, 0, 0, 4, 1, 1, 4, 4, 0, 4, 2, 0, 4, 3, 4, 4, 4, 2, 0, 0, 0, 3, 2, 4, 4, 4, 4, 0, 2, 0, 0, 4, 2, 4, 3, 0, 4, 4, 2, 1, 4, 4, 3, 3, 2, 4, 2, 4, 4, 3, 3, 0, 4, 1, 2, 4, 0, 0, 0, 4, 4, 4, 4, 4, 2]

7.2. Features Extracted from the Annotation

The annotation process is essentially creating an attributed directed graph: a network of the characters in a plot summary and the relationships between them. The feature extraction task is therefore to create a vector space embedding of graph features. Following Gibert 2012, local graph features were selected to approximate the network's broader structure. Gibert's method is:

"[...] to associate feature vectors to graphs in a simple and very efficient way by just putting attention on the labelling information that graphs store. In particular, we count frequencies of node labels and of edges between labels. [Despite] their locality, these features are able to robustly represent structurally global properties of graphs, when considered together in the form of a vector."

However, Gibert studied non-directed graphs with single labels on nodes and edges. The gold standard graph contains multiple labels on both the nodes and the edges in addition to having directed edges. Rather than getting highly specific, non-overlapping counts for Entity labels, counts are collected for not only each Entity node label in the gold standard graph, but also for each pair of a node's label and its various attributes such as subtype and plurality. This leads to overlapping counts: a node with label L and attributes A_1, ... A_n will add to the counts of n+1 dimensions in the vector instead of to just one dimension. This pads the vector and serves to localize the network features being embedded.

Similarly, a great deal of care needs to be taken in selecting how the edges, or Relations, are vectorized. All combinations of attributes of all combinations of the starting node or agent, the target node or patient, and relationship trigger

word could be taken into account; but this would produce an exponentially long and unusably sparse vector. Alternatively, only the Relation's own attributes might be counted, resulting in $5 \cdot 5 = 25$ additional dimensions for the vector, more like Gibert's approach which counts only individual edge labels. As the edges of the gold standard make up the most crucial pieces of information in the task of describing relationships, the best approach for this application is likely somewhere in the middle, including at most one or two labels for a given edge and not searching for information on any path beyond its endpoints.

The current implementation counts relationships overall, as well as those in which protagonists, antagonists, and other actors cause the relation to exist. It also counts those in which each of the above does something to help themselves, others, or no one; and those in which each does something to harm themselves, others, and no one. Furthermore, it counts those in which an object is the source of the relationship, as well as when these relationships benefit or harm an antagonist, a protagonist, another character, or no one.

This set of counts generates a less sparse vector with more local information through overlap and double-counting. For example, in a mutually beneficial relationship, a relation connecting vertices with labels A and B will add a count to at least two dimensions: from-A-benefits-self and from-A-benefits-others. It is also more sensitive to situations caused by objects; instead of considering the same range of relationships as with characters, only the labels of the affected parties are relevant.

This vector also highlights the difference between self-service and mutually beneficial relationships as well as neutral and self-harmful ones, as opposed to paying closer attention to the extreme particulars of who is doing what to whom. It reflects which types of characters are affected by a relationship and in what ways, along with some of the details of the agency of the relationship. It doesn't capture all of the details: it leaves the type of relationship to the local trigger counts rather than incorporating them here. Certain combinations are underspecified if they are not likely to be relevant–for example an object benefitting itself. It emphasizes characters' actions while mildly de-emphasizing things that happen to them. The number of dimensions, 31, is much reduced from the thousands that could arise from capturing every detail of the edge.

8. Machine Learning Experiment Results

Five means were used in the clustering algorithm. The algorithm was built using the NLTK machine learning module for k-means clustering, taking the most-assigned cluster out of nine attempts for each document in the gold standard.

ann_clusters = [3, 3, 4, 2, 2, 3, 4, 4, 1, 2, 2, 2, 3, 0, 4, 2, 4, 1, 0, 2, 2, 1, 2, 0, 0, 1, 2, 0, 2, 1, 2, 4, 2, 2, 2, 0, 4, 4, 1, 4, 4, 1, 3, 3, 1, 4, 4, 4, 2, 2, 4, 3, 3, 4, 3, 0, 2, 2, 2, 4, 2, 0, 3, 4, 1, 2, 2, 0, 1, 3, 2, 4, 4, 1, 2, 2, 2, 3, 1, 1, 2, 1, 1, 3, 4, 1, 3, 0, 4, 0, 4, 1, 2, 4, 1, 2, 1, 4, 0, 0, 1, 4, 2, 4, 2, 3, 4, 1, 1, 1, 1, 4, 4, 1, 1, 2, 3, 0, 1, 2, 4, 1, 1, 4, 1, 4, 1, 1, 0, 1, 1, 2, 2, 1, 1, 0]

Evaluation was attempted by choosing a plot summary from the corpus at random and having a potential reader rate his or her interest in it from 1 (highly disinterested) to 5 (highly interested). The reader then rated a different summary chosen at random from the same cluster as the first; one from another cluster; one from the same cluster as generated by unigram features; and one from outside the original's cluster based on unigram features. The results may be seen in the figure below. Each column is a measure of ratings.

Init. text	in-clus.	out-clus.	BOW in	BOW out-clus.
1	3	2	3	2
2	1	1	1	4
3	1	2	1	4
1	1	1	1	2
1	1	1	2	1
1	1	1	1	2
3	1	2	3.5	1.5

Table 1: Reader Ratings

9. Conclusion

The groupings generated by k-means clustering were stable across multiple iterations of the clustering algorithm. They are markedly different from the clusters created from binarized unigram or "bag of words" feature vectors.

In the tests in which raters gave a rating of interest in books based on the summaries in the corpus, significant ceiling effects were seen due to the fact that many readers are generally uninterested in the genre selected. Those who did not tend to rate every summary with extreme disinterest showed a pattern toward rating same-cluster summaries more similarly to one another than to different-cluster summaries. This trend indicates a degree of success over the baseline. More rigorous evaluation is warranted to confirm that this trend is correct, stable, and statistically significant. Many factors aside from character interactions likely play a role in reader interest, and a fully-implemented recommendation system will attempt to incorporate these factors by adding topic analysis, increasing the number of clusters and the corpus size, and improving the annotation specification.

10. Bibliographical References

Gilbert, Jaume. (2012). Vector Space Embedding of Graphs via Statistics of Labelling Information. UAB Barcelona.