SentNet: The Effects of Sentiment on Information Propagation

Jan Overgoor
Stanford University
overgoor@stanford.edu

Evan Rosen
Stanford University
emrosen@stanford.edu

Abstract

We aim to model online news as an evolving self-replicating phenomenon under selection pressures due to the sentiment which it evokes. We explore this idea by applying lexicon-based sentiment classification to clusters of quotes created using the MemeTracker framework. We perform correlation tests on the popularity, sentiment and network features extracted from each cluster and found that sentiment features are more tightly correlated with future popularity than features of current popularity are.

1 Introduction

We take a view of language much like the childhood game of telephone. An event originates at some point in a community of speakers and listeners (or readers and writers) which then gets passed along through a noisy network of interactions. Over time, features of these interactions may begin to influence the very content of the language that exists over such a network. Alternatively, structural features of that network may also have an effect on this content, different points of origination can for example provide a greater chance of diffusion. Our goal is to explore the effect which these selection pressures have, if any, upon the language itself, through tracing the success and failure of specific *memes* through such a network.

An interest in these sorts of general information dynamics is not new, of course. There has been widespread sociological interest in the practice of story telling, from its traditional folk roots to the modern incarnation of Internet urban legends. Over the past several decades, a shift in this literature has taken place in which the success of a story is explained less in terms of the circumstances in which it arises but more as function of intrinsic features of the story (Heath et al., 2001). On this view, it is an open question as to exactly which intrinsic features contribute to the propagation of a story. For example, it is common to distinguish between the informational and emotional content. However, the distinction remains largely theoretical. In one of the first empirical papers to address this question, Heath (Heath et al., 2001) take an important step in leveraging the scale of data available on the web to evaluate the role of disgust in urban legend propagation.

(Heath et al., 2001) score a set of 100 urban legends according to the number of distinct motifs of disgust that are present, and then test how often each story appears across a number of online urban legend aggregators. While they find that there is in fact a strong correlation between disgust and popularity in urban legends, their approach does not clearly generalize to a broader view of storytelling. For example, the hand annotation of stories using a hand-built coding system for emotional motifs is not scalable, nor can one rely upon highly structured resources such as urban legend aggregators for all story domains. We hope to address both of these problems through the use of automatically generated sentiment lexicons, a large web crawler corpus of online news and blogs, and information retrieval techniques.

We also wish to go one step further in casting news stories and blog posts as evolving entities of this sort. News and blog authors are also news and blog readers, and it seems inescapable that the content that they choose to write on should be related to that which they read. But of all the content that they have encountered, which topics do they choose to write on? What stories will take off? We investigate the role of sentiment and subjectivity in this process, as well as the structure of the informational network that produced the story. Thus we can formalize our problem concisely as an investigation of the correlations among three variables:

sentiment, network structure, and popularity.

2 MemeTracker

In order to evaluate the popularity of a news story, we used the dataset and framework described in (Leskovec et al., 2009). The dataset consists of a large corpus of blogs collected through the Spinn3r API [1] , a blog RSS feed aggregator. Each entry records the URL, the page title, the date of publication, any outgoing hyperlinks, any quotations, and the page content itself. The date of publication, in this context, corresponds to the time at which the page was pushed to an RSS feed, which provides a level of temporal granularity of an hour. The Spinn3r fire hose corresponds to roughly 1 million pages and 0.5 million quotes per hour.

We also used the same approach for tracking memes. While the question of how to resolve whether two texts are *about the same thing* is far from solved, Leskovec et al. (2009) showed that newswire can be efficiently checked for similarity by checking the quotes which it contains. Thus we can implicitly represent an item of news by the quotes with which it is associated and track the popularity of that item by the number of distinct pages on which a particular quote appears. To further improve upon this intuition, Leskovec et al. (2009) also take care to resolve textual variants of a particular quote so that subquotes, spelling errors, and even mis-rememberings all count as instances of that same quote or news story. In our work, these quote clusters take on the additional role of normalizing for informational content.

To build these clusters, we first construct a weighted directed graph over all of the quotes in the corpus where an edge exists from quote q_1 to quote q_2 in the case that there is an overlap of length greater than 10 or an edit distance no greater than 1, and $len(q_1) < len(q_2)$. The weight for each edge is inversely related to the edit distance or length of non-overlapping text, and is directly related to the frequency of the longer quote (to encourage the inclusion of subquotes in more popular superquotes). Using inverted indices and optimized c code, we can construct the quote graph corresponding to 25 hours of news in about 1 hour.

Next, we partition this graph into connected components corresponding to quote clusters in a way which attempts to nest each quote under the most likely quote of greater length, from which it was excerpted (in the case that there is such a candidate quote). This corresponds to the problem of creating a set of singly rooted DAGs [2], while removing the fewest edges. While Leskovec et al. (2009) show that this problem is NP-complete, they also provide a greedy approximation. Their algorithm proceeds from any initial root nodes (out-degree is zero) through the graph, removing all of the outgoing edges from a given quote which do not connect to the most strongly connected connect cluster. This has the desired outcome of leaving each quote as a member of only one cluster.

3 Sentiment Classification

There are many approaches to representing the sentiment contained in a text, with little available knowledge as of yet about their relative merits. We take a strictly lexical approach, representing the sentiment of a phrase as a function of the words it contains, each of which is assigned a word level sentiment value. We treat the sentiment of an individual word in the traditional one dimensional sense where a term can either be positive or negative and has a correspondingly positive or negative real value. While this distinction is rather coarse, it serves to answer our general question about whether the latent dimensions of emotion might play a role in the success of online news. We explore three different versions of this approach based on different ways to build these sentiment lexicons.

Due to the difficulty of formulation a generic definition of positive and negative sentiment, we take a common approach of using a small seed set for which we know the sentiment polarity and propagating those polarities through a lexical similarity graph, which we hope also encodes sentiment similarity. We used the same starting set of positive and negative terms, from (Turney and Littman, 2003), to seed both methods. The first approach, based on a description by Godbole et al. (2007), uses WordNet (Miller, 1995) to find terms related to those in the seed set. Every iteration expands the current set with the synonyms and antonyms of the terms in the current set and assigns, with a decay factor (we used 0.9), respectively the same or the opposite value to the newly found words. The final score of a word is the sum

[1] `spinn3r.com`

[2] (Leskovec et al., 2009) define a singly rooted DAG to be a graph where there is only one node with out-degree of zero.

of all the incoming paths, only considering those paths that have a maximum number of ‘polarity flips (we used 2), when the current value is different than the one started from.

While WordNet based propagation algorithms have the advantage of leveraging human annotated synonymy relations, they may fail to generalize in the messy and unstructured domain of online content. To address this limitation we experimented with a similar propagation algorithm that substitutes a notion of webpage co-occurrence for WordNet synonymy. The algorithm, described in (Velikovich et al., 2010) creates a context vector for each word consisting of the number of times it cooccurs on the same web page. A graph is then constructed over the vocabulary such that there is an edge between any two words for which the cosine similarity score of their context vectors is greater than 0.01. Positive and negative sentiment is then propagated outwards from seed sets over these edges according to their weights using a modified version of label propagation.

While this algorithm is very appealing in principle, it posed too great a computational barrier to be especially useful for this project. The $O(n^3)$ computation involved in computing the cosine similarity scores for all pairs of nodes forced us to limit the lexicon to around 5k words. To make the best of this restriction we made sure that our co-occurrence counts were meaningful by taking only the 5k most common words from a large corpus spanning an entire month (300GB of text). Though it is difficult to call this technique a success, the benefits of such an approach did begin to show themselves. For example, many of our web pages and quotes were actually not in English. While this presents in insurmountable problem to the WordNet lexicon, by using the text from quotations from the web as the inputs to the web propagation algorithm, we were guaranteed to provide sentiment scores (though likely inaccurate) for common words, regardless of language. On the whole however, we were not able to fully explore this approach due to time constraints.

To check whether we had correctly implemented these methods and to provide something like a baseline, we also included the widely used SentiWordNet lexicon (Baccianella et al., 2010). This lexicon provides positive and negative scores for each synset in WordNet, however we needed to combine these into a single real-valued score. To do this, we used an approach inspired by (Velikovich et al., 2010) where we first normalize each positive and negative score, by the sum of all scores for that polarity. Then we just subtract the normalized negative score from the normalized positive score. Finally we removed any terms with a score of zero.

The WordNet lexicon had a size of 9,949, the web propagation lexicon had a size of 6,269 and the SentiWordNet lexicon had a size of 30,978. To make the scores provided by these approaches comparable, we also z-score normalized each polarity score. In Table 1 are displayed the top and bottom valued words for comparison.

WordNed		**WebProp**		**SentiWordNet**	
good	44.03	correct	0.38	top-hole	3.18
superior	16.28	re	0.35	admirability	3.18
dependable	16.01	magazine	0.34	first-class	3.18
reliable	11.17	gaza	0.33	top-flight	3.18
best	10.90	israel	0.33	wonderfulness	3.18
...	...	...	...	...	...
unfavorable	-11.24	magic	-0.64	sooty	-1.19
inferior	-11.66	96	-1.12	understock	-1.19
worst	-12.64	harman	-1.22	seldom	-1.19
wrong	-14.65	harriet	-1.22	jagannath	-1.19
evil	-22.10	apologised	-1.27	underachieve	-1.19

Table 1: The top 5 most positive and negative entries in the used sentiment lexicons.

Our actual sentiment scoring function scores a text by taking the average over the sentiment scores of the words that have a score (e.g. that are present in the dictionary). Following results reported by (Pang and Lee, 2008), we only considered the feature *presence*, in contrast to the convention in information retrieval which uses feature *frequency* instead. Another possible sentiment measure is that of *subjectivity*, which is calculated in a similar way, except for that it takes the absolute values of the sentiment scores.

4 Network features

Because the content we use comes from linked webpages, we can also use their interconnectedness as a feature that says something about the nature of the information that is presented. Linking to another webpage is a common way to cite, support or criticize something and provides a definite semantic connection between the two documents. A webpages connectivity also provides a different perspective on its popularity. To use this we look at the set of webpages as a network, where each node represents a page and a hyperlink a directed edge to the node linked to. The topology of the resulting network has many possible applications, but we primarily use it to generate features to correlate with prevalent quote variants.

The *connectedness* of the network can be expressed by taking the number of edges, normalized by the number of nodes in the graph. When the considered graph is seen as embedded in the larger webgraph of all pages, we can take the total number of *in degrees* of the nodes in the graph, again normalized by the number of nodes, as an indicator of popularity. The commonly used model of preferential attachment for the power law growth of networks refers to this value and has characteristics of an evolutionary process. The actual *growth rate* of the network (number of new nodes / time span) is another possibly interesting feature. Finally there is the feature of *average path length*, which is the average over the number of steps it takes to traverse between an arbitrary node in the network to another one.

In addition to the network that consists of the documents containing relevant quotes, we also look at its extended networks that contains the documents linking to and linked to the original document set. Every discussed feature can be taken from the original network or its extension, which is denoted by F^E.

5 Methodology

In order to find structure in the data, we cast the problem as a prediction task as follows: each quote cluster has a certain lifespan, with a corresponding popularity function, mapping a time point to the relative or absolute popularity of that quote on that time. For each quote cluster C_i we take a time point t_i^* such that $t_i^* = (t_i^{end} - t_i^{start}) \cdot 0.2$, where the time is measured in hours. This provides two perspectives on a cluster: the state of the cluster at time t^* (C_i^*) and the state of the cluster over its entire duration, with the end of the data sample as the global end. For both perspectives, all discussed features can be considered: the quote tokens in C_i^* are those from C_i that occur on webpages published *before* t_i^*.

We aim to find significant dependencies between the early C_i^* values and their C_i outcomes. For this we applied the Pearson correlation coefficient on the features we extracted[3]. This measure computes linear correlations, using a Students t distribution. In addition to this we calculate p-values for the correlations, which represent the probability that there is no correlation between features. This creates a confidence interval of the found correlations.

We group our features in the three aforementioned groups of popularity, sentiment and network to evaluate the separate hypotheses of structural relations between the groups. A list of the used features with their descriptions is given in the Appendix. The main correlation we want to test is between sentiment features at t^* and the general popularity features. On the same level there is also the relation with the networks features, and in general the correlations between all three types are interesting, but here we focus on just sentiment and popularity.

Correlation values alone do not provide enough basis for accepting or rejecting hypotheses, but they can be used for selecting useful features. Also, there is no clear way to set a p-value threshold to distinguish good feature pairs. To get some perspective on acceptable p-values for this task, we compare them with a baseline correlation between the popularity at t^* and the overall popularity, on the view that if we know nothing about dynamics of the news, we can at least use the current popularity of a news items as a predictor of its future popularity. A cluster will never lose cumulative popularity over time. So, by guessing the current popularity we have at least ruled out an impossible set of circumstances.

For our experiment we used a selected corpus that we compiled from the data provided by Spinn3r. Given that we wanted to consider as much English content as possible we collected the content published between 07:00 and 13:00 ET, the days from 10/21/2010 to 10/30/2010. We chose these days on the assumption that the US elections would populate the news with recurring quotes.

With this same framework we can probe even deeper in the question about how sentiment affects the news. Though in some cases, quotes might be manufactured to evoke a stronger emotional reaction, for the most part in the case of news quotes, the space of legitimate quote variants is limited to actual substrings. Thus, in much the same way biological evolution occurs, the space of variants is limited. When we wish to demonstrate the exis-

[3] Implementation in MATLAB : http://www.mathworks.com/products/matlab/

tence of an evolutionary selection pressure, we do not expect of every organism that it be on a straight path towards optimizing itself with respect to that pressure. Rather, we expect the current variant that is best optimized with respect to that pressure, to demonstrate greater fitness than its most similar variants. The analogy to our case is that most sentiment rich quote variants *within* a cluster should be the most successful. We test this hypothesis using a similar correlation framework and give the most strongly correlated features at time t^* for features which are computed for a single token rather than cluster.

6 Results

An essential component of our system is the effective identification of quote variants. These not only allow us to generalize our claim about the role of sentiment to a broader notion of stories, but also to acquire a more granular picture of how selection might operate. To test our implementation of the MemeTracker quote clustering technique we printed out the quotes that had the largest number of distinct types. As we can see in Figure 1 the quote matching appears to be working in the way we would like. Quote 2 is indeed a subquote of quote 1, and is not merely a spurious coincidence of terms. This cluster also exemplifies a common pattern in the data where a single exhaustive quote subsumes a large number of shorter and often less accurate quotes.

Using these quote clusters, we would then like show that the sentiment of those clusters effects which quote cluster and news stories will become popular. To see this popularizing in action is the most convincing proof. However, we can preliminarily ask whether the general composition of popular news is consistent with this hypothesis. If sentiment has an effect on popularity it should also be reflected in any snapshot of quote cluster popularity. To answer this we start by plotting the distribution of sentiment scores across types in Figure 2. If we imagine that each quote only appeared for the first time once, then we can view this plot as the initial sentiment of online news and blogs before any duplication has occurred. This figure shows that we have a very sharply peaked distribution around sentiment, $s = 0$ which is roughly what we might expect: most quotes are of neutral content. To then see how these quotes fare over time with respect to sentiment, we can take the final token count for quotes with a sentiment score s and normalize by the number of quote types with that sentiment score. This then shows us the average popularity for each quote by sentiment score. As we can see in Figure 3 this distribution is not simply flat. Instead, the quotes with more extreme positive and negative values are more popular per quote type up until a certain degree of polarity.

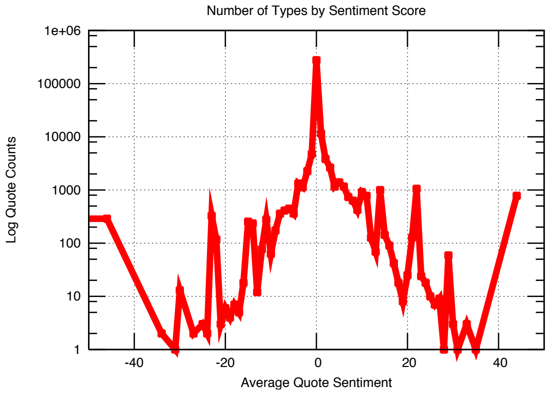

Figure 2: Histogram of Quotes by Average Sentiment Score

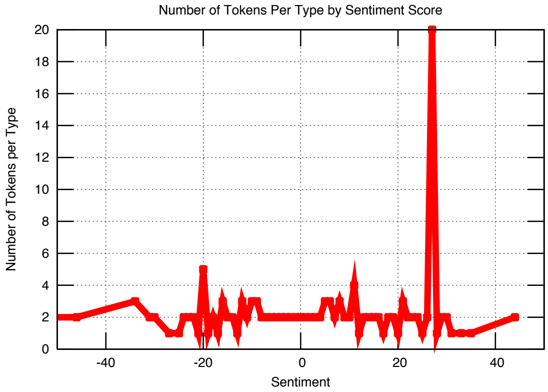

Figure 3: Sentiment vs. Average Popularity Per Cluster

6.1 Cluster-level Correlations

Tables 2, 3 and 4 display the top 5 resulting correlation values and p-values, sorted on ascending p-value. A glossary of the feature names is given in Appendix A. Table 2 portrays the correlation between popularity at time t^* and the overall popularity, expressed by number of pages. As can be seen, the only correlating feature are NPages$_t$ and NTypes$_t$, which is because these was the only t^* popularity feature we had. NPages$_t$, not surprisingly, is strongly correlated to its future projection, to an almost logical relationship

Table 3 displays the top correlations between

1. (count=47,sent=-0.006560): you are swine the children that you bear from this marriage will all be bastard swine your marriage is not a valid one you are not the kind of people who can have a valid marriage one of you is an infidel the other too is an infidel and we have reason to believe an atheist who does not even believe in an infidel religion
2. (count=1,sent=-0.009942): you are swine the children that you bear from this marriage will all be bastard swine your marriage is not a valid one
3. (count=5,sent=-0.011484): the children that you bear from this marriage will all be bastard swine
4. ...

Figure 1: Quotes in an example cluster

From	To	Corr	P Value
NPages$_t$	NPages	0.0942	0.0004
NTypes$_t$	NPages	0.0131	0.6255

Table 2: Correlation between features of clusters from Popularity$_t$ to Popularity. F_t is F before time t^*.

From	To	Corr	P Value
SentConWp$_t$	NToks	0.0541	0.0443
SubjConSw$_t$	NPages	-0.0464	0.0842
SentConSw$_t$	NToks	0.0464	0.0844
SubjConWp$_t$	NPages	-0.0425	0.1141
SentConGd$_t$	NTypes	-0.0392	0.1448

Table 3: Top 5 correlations between features of clusters from Sentiment$_t$ to Popularity. F_t is F before time t^*.

sentiment at time t^* and the overall popularity. SentConWp$_t$ performs really well on its p-score, but all three lexicons are represented. One interesting pattern is that the *sentiment* features have a positive correlation with popularity, while the *subjectivity* features are negatively correlated. Note also that the features for the sentiment of the types are completely absent, suggesting that the sentiment expressed in the quote itself is not a good indicator for future success.

Table 4 contains the correlations between network values at time t_t and the overall popularity. The p-values are generally very low and the correlation is positive, suggesting that a tighter network structure is a good indicator for future popularity. These values are, however, based on less training data as not all pre-t^* clusters have a network to speak of.

From	To	Corr	P Value
GrowthRate$_t^E$	NToks	0.0801	0.0029
Edges$_t^E$	NToks	0.0726	0.0069
InDeg$_t^E$	NTypes	0.0633	0.0186
Edges$_t^E$	NTypes	0.0566	0.0352
GrowthRate$_t$	NPages	-0.0494	0.0660

Table 4: Top 5 correlations between features of clusters from Network$_t$ to Popularity. F_t is F before time t^*, E is the extended network.

We also looked briefly at the correlations between the different features at time t^* on the one hand and the sentiment and network features of the whole cluster at the other. The results seemed to indicate a consistent negative correlation between sentiment and network features. This might also be an interesting research direction. Other correlations generally produced p-values of 0.

In addition to the correlation analysis, we ran a couple of tests with the decision tree implementations in WEKA[4] to see if our features at time t^* can predict values for the overall popularity. The amount of available time and space restricts an extensive evaluation, but the early results are promising. For the DecisionStump algorithm, a single layer decision tree, SubjConGd$_t$ was found to be the strongest indicating value. In addition to this, the first three rules of the tree resulting from the multilayer tree REPTree depended on sentiment features. The second tree improved performance on the error values with a very small margin. These last results indicate that sentiment features might indeed be used to predict popularity values.

[4] Data mining software at http://www.cs.waikato.ac.nz/ml/weka/

From	To	Corr	P Value
NPages_t	NRelToks	-0.0112	0.7717

Table 5: Correlation between features of types from Popularity_t to Popularity. F_t is F before time t^*.

From	To	Corr	P Value
SubjConSw_t	NToks	-0.1176	0.0023
SentConSw_t	NPages	0.0963	0.0128
SubjConWp_t	NPages	-0.0735	0.0575
SubjConWp_t	TimeSpan	-0.0682	0.0782
SentConWp_t	TimeSpan	0.0577	0.1363
SubjConWp_t	NRelToks	-0.0507	0.1906
SentConWp_t	NRelToks	0.0430	0.2668

Table 6: Top 5 between features of types from Sentiment_t to Popularity. F_t is F before time t^*.

6.2 Type-level Correlations

As discussed in our methodology, we also have the ability to take a more fine-grained view of the selection pressures operating on online news by considering an individual quote type as the unit of analysis. With respect to the larger picture of how news evolves, these may be thought of as strains of particular news story. We give a parallel set of results below, summarizing the strongest correlations between the features of single quotes type at time t^* and features for that same quote during the remainder of its lifespan.

For the between-type feature of Popularity_t (Table 5) we found only one meaningful correlation.

To return to our discussion of within cluster variation, Table 3 shows that if we wish to predict which quote variant will be most successful within a cluster, we should look to the sentiment of the quote variants. The sentiment and subjectivity of the context scored with the web propagation lexicon are both correlated with the NRelToks feature, which represents the proportion of tokens which a particular quote constitutes in a cluster. If we look closer at the sign of the correlate, however, things become less clear. Sentiment is positively correlated with the relative number of tokens, while subjectivity is negatively correlated.

Our initial expectation was that if sentiment is positively or negatively correlated with any feature, then subjectivity should be positively correlated. However subjectivity, which is just the absolute value of sentiment score, is negatively correlated. Our best explanation of this result is that a positive correlation with sentiment does not entail that a feature co-occurs with positively scored quotes, but only that *more* positive or equivalently, *less* negative, quotes tend to co-occur with quotes which are more popular within their cluster. This suggests that the real relationship between sentiment and relative number of tokens, might rest in a failure of strongly negative quotes to become especially popular. This would create a positive correlation between sentiment and relative number of tokens, and also create a negative correlation between subjectivity and relative number of tokens, for as the the sentiment scores become less negative, they become closer to zero which implies a smaller subjectivity score.

To give this line of reasoning a concrete example, let us revisit the example in Figure 1. Our hypothesis can be applied here to predict that the more sentiment rich quote will out compete its variants. Though we do not simply see that the quote with the highest sentiment score is the most popular, it is not in contradiction to our theory. It is likely that not all of the quotes are of the same age, perhaps giving quote 1 a head start. What is important here is to show how different subquotes can have different scores even though they are ultimately just excerpts from the same single text. This is possible because our sentiment score is normalized by the number of tokens, so that shorter quotes can have higher sentiment than longer quotes in which they are subsumed. Thus we can imagine how the rise of one quote variant within a cluster might demonstrate something about sentiment. We can imagine that if quotes 2 and 3 were created simultaneously, quote 2 might do better, because it is not carrying the dead weight of a redundant instance of "swine and the rather dry claim that your marriage is not a valid one.

It is also worth noting the strong correlation between sentiment and time span. In the same way that the fitness of an organism is a function both of its fecundity and its longevity, we might expect the same features which affect the popularity of a story to affect the period of time over which people are interested in that story. The same picture which we encountered in explaining the correlations between sentiment and relative number of tokens also applies here: the proper interpretation of a negative correlation with sentiment and a pos-

From	To	Corr	P Value
AvgPathLen$_t^E$	NToks	0.1007	0.0092
Connectedness$_t^E$	NToks	0.0532	0.1697
InDeg$_t$	NToks	0.0358	0.3552
GrowthRate$_t^E$	NRelToks	-0.0194	0.6168
AvgPathLen$_t$	NRelToks	0.0189	0.6256

Table 7: Top 5 between features of types from Network$_t$ to Popularity. F_t is F before time t^*, E is the extended network.

itive correlation with subjectivity is that the less negative a story, the longer it will be discussed.

In general, the sentiment features correlate better with future popularity than the baseline of popularity itself. One problem with this conclusion is that the comparison with the baseline might not be fair as it assumes that the relation is linear, while there is evidence that social networks often grow in a non-linear fashion (Clauset et al., 2009).

Finally, Table 7 displays the correlation between the network features on type level to the final popularity. The p-values are again quite good and, as in Table 4, the correlations are almost exclusively positive. Only growth rate displays a negative correlation, which is somewhat counter-intuitive. Compared to Table 4 the results are, however, slightly less obvious because of the presence of the AvgPathLen$_t$ feature. The suggestion is that a longer average path length will lead to more popularity.

7 Conclusion

In general, the behavior of the different features is not conclusive enough for strong conclusions, but some interesting patterns came up. Sentiment features, especially *subjectivity*, turn out to be relatively strongly correlated with popularity statistics, even though the correlation is negative. The network features, on the contrary, do not perform well. This is mainly because of the sparsity of the network data - the networks in the corpus we used were not densely connected. This might also have something to do with out implementation. A possible extension into this last field would be to assign an edge between nodes when they quote the same phrase, which would populate the the graph more.

Overall, the combination of sentiment analysis and the MemeTracker framework provides a wide scope of future research directions. It is a rich trove of aspects of information propagation that can be investigated and compared. To start, we only used very basic techniques for the sentiment analysis. This field has itself only just started to develop and there are many possible approaches to the general problem of sentiment extraction. Starting points include to use n-grams as opposed to single terms, or for example POS tags to disambiguate.

Another possibility for elaboration is the application of machine learning to find structure in the data we gathered. As we discussed, the C_i^* and C_i pairs can be used a a training set, so that the trained system can try to predict C_i values from C_i^* input. An example system that does this would take the current moment as t^*, the recent published (clustered) content as C_i^*'s, and predict which news stories and/or quote variants will become the most popular in the future.

In this paper we only looked at the competition between clusters themselves. Which news flash will become the next big story? However, the inquiry into which quote variant will prevail within a cluster is also an interesting research direction. This is a much more subtle enterprise which might reveal

References

S Baccianella, A Esuli, and F Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. volume 25, pages 2200–2204.

A Clauset, CR Shalizi, and MEJ Newman. 2009. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703.

N Godbole, M Srinivasaiah, and S Skiena. 2007. Large-scale sentiment analysis for news and blogs. *Proceedings of the International Conference on Weblogs and Social Media (ICWSM).*

C Heath, C Bell, and E Steinberg. 2001. Emotional selection in memes: The case of urban legends. *Journal of Personality*, 81(6):1028–1041.

J Leskovec, L Backstrom, and J Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506.

GA Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):41.

B Pang and L Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

P Turney and M Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346.

L Velikovich, S Blair-Goldensohn, K Hannan, and R McDonald. 2010. The viability of web-derived polarity lexicons. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–785.

Appendix A

Here is a complete list of the features we used. All networks features F can be called as F^E, using the extended network. All features F can be called as F_t, taking into account only the occurrences before time t^*.

Popularity features

NTypes number of unique quotes in the cluster

NToks number of quote instances in the cluster

NRelToks number of quote instances in the cluster, normalized by the size of its superset

NPages number of webpages where quotes occur

TimeSpan time interval in hour between first and last occurrences of instances of the cluster

Sentiment features

SentConGd sentiment of quote context based on the WordNet lexicon

SentTypeGd sentiment of a quote based on the WordNet lexicon

SentConWp sentiment of quote context based on the web propagation lexicon

SentTypeWp sentiment of a context based on the web propagation lexicon

SentConSw sentiment of quote context based on SentiWordNet

SentTypeSw sentiment of a context based on SentiWordNet

SubjConGd subjectivity of quote context based on the WordNet lexicon

SubjTypeGd subjectivity of a quote based on the WordNet lexicon

SubjConWp subjectivity of quote context based on the web propagation lexicon

SubjTypeWp subjectivity of a quote based on the web propagation lexicon

SubjConWp subjectivity of quote context based on SentiWordNet

SubjTypeWp subjectivity of a quote based on SentiWordNet

Network features

Nodes number of pages in the graph (=NPages)

Edges number of edges (URL's) between nodes

Connectedness number of edges normalized by number of nodes

InDeg total number of incoming edges to nodes in the graph

GrowthRate number of new notes normalized by difference in time

AvgPathLen average path length (number of edges required to get from one node to another) between nodes in the graph