News Detection in the Blogosphere: Two Approaches Based on Structure and Content Analysis

Leandro Franco
School of Computer and Communication Sciences,
EPFL, Switzerland
leandro.franco@epfl.ch

Hideki Kawai
NEC C&C Innovation Research Laboratories,
8916-47, Takayama-cho, Ikoma, Nara, Japan
h-kawai@ab.jp.nec.com

Abstract

In this paper, we study a subset of the blogosphere created by *spinn3r* during August and September 2008 containing 20.5 million posts. We propose two approaches to detect and filter important news and events published in blogs. The first involves exploring the structural properties of the post network and the information cascades within it. For the second approach, we use a scalable algorithm to analyze the content in posts and cluster them accordingly. In both cases, we use the idea of diversity to develop simple mechanisms able to differentiate interesting news from the rest of the blogosphere (such as spam, advertising and politically extreme discussions).

1. Introduction

The Blogosphere gave a new dimension to online expression. It is true that users could create websites long before creating weblogs (or blogs) and it is true that a weblog can be seen as just a website. But the platforms supporting bloggers became so easy and intuitive that anybody could start blogging. In addition to this information explosion, blogs have certain characteristics that differentiate them from normal websites. Every blog consists of a number of entries (or posts), which are made up of a title, body and publishing date. Although adding the date is a minor technical detail, it will prove incredibly useful to analyze the temporal behavior of blogs and the people behind them. Another important detail from weblogs is that entries contain links pointing to other blogs or even posts within the blogs. This is particularly useful if we are interested in creating a blog graph to explore social connections or if we would like to create a post graph to analyze the relationship between subject or ideas.

This paper is based on a dataset provided by the spinn3r company (Burton 2009). There are 44 million posts taken during August and September 2008 and comprise different blog domains and other sources. From this dataset, 22 million posts are classified in the 13 tiers created by spinn3r and the rest are left in a notier group. In this paper, we will only use those posts successfully classified instead of the whole dataset. Furthermore, we will restrict ourselves to the 11 million posts classified as English when performing content analysis.

In the first part of this paper, we focus on the post graph and on how interesting news or topics trigger information cascades within it. To create this graph we extract all the links in the body of posts and check if they were pointing to another post within the dataset, if they do, we create a directed edge between those two posts. We analyze this graph, point out some of its structural properties and extract the information cascades within the graph (a connected subgraph with an identifiable root). We show some interesting properties about information cascades, the most significant being the relationship between the shape of a cascade and the interest of the general public on it. Star-shaped cascades tend to be interesting news while chain-like cascades tend to be spam or tightly coupled communities of very narrow interests (extreme political issues). Based on this notion we are able to filter out most of the irrelevant cascades, although on some occasions we leave out interesting ones as well.

In the second part, we try to find out cascades given by content. For this, we design an algorithm for content/post clustering inspired by the online bin packing problem (Vazirani 2001). The algorithm is scalable if we allow pruning at each level. It can be run in parallel since we split the dataset in blocks and analyze them individually. Once a result is obtained with the algorithm, it can be used to incrementally add more posts to the existing clusters or even to create new clusters. After finishing the algorithm and obtaining the set of clusters, we proceed to analyze their *diversity* and classify them according to this. In an analogous way to cascades, we propose a simple heuristic to filter out noninteresting news (in this case we find mostly advertising). At the end of the paper, we also give a graphical visualization representing some of the biggest clusters found by our algorithm.

2. Previous Work

One of the closest topics to the approach taken in the first part of this paper is the analysis of cascading behavior on the internet done by Leskovec and colleagues. In (Leskovec 2006) they analyzed cascades in a recommendation network while in (Leskovec 2007) they did the same for a blog network. In that last paper, they concluded that blog posts do not have a bursty behavior but instead behave cyclically (weekly), and they proposed a simple epidemiological

model that seems to emulate well the properties of real cascades.

Recent work performed by Leskovec et al. does not focus on the explicit structure of blog connections, instead they track the actual content of posts. The "MemeTracker" (Leskovec 2009) will look for quoted phrases among all posts and will cluster them by proximity. This approach gives a clear idea of what is most popular at any point in time and is related to the study we do in the second part of our paper, where we want to extract the analogous of cascades when talking about content instead of structure.

Another recent study (Cha 2009) uses the spinn3r dataset to analyze the diffusion of media content through the blogosphere (mostly youtube videos). In this paper they mentioned two main patterns of diffusion: flash floods and ripples. This means some videos become very popular in a short amount of time (political news) but also vanish sooner while others achieve popularity over a long period of time.

Earlier work includes (Gruhl 2004), (Adar 2004) and (Adamic 2004) among others. In (Gruhl 2004), the authors used term frequency to find out important topics or categories of diffusion, in (Adar 2004), the main insight was that blogs are not only related explicitly (through links) but also implicitly through content, and in (Adamic 2004), they give a good example of practical applications of blog analysis. In their paper they established similarities and differences between blogs with different political inclinations.

3. Dataset

Spinn3r is a company for web indexing focused on the blogosphere, they claim to "provide raw access to every blog post being published - in real time". This company handed over a dataset for the 3rd & 4th *Int'l AAAI Conf. on Weblogs and Social Media* and encouraged researchers to participate in a data challenge held in conjunction with the conference. The dataset contains 44 million blogs and its total size is around 142GB uncompressed, the posts were collected during August and September 2008 and are presented on a basic XML format indicating title, source, body and other properties. All the posts in the dataset are classified in tiers according to their relevance. There are 13 of these tiers plus a general tier *nogroup* containing all posts that could not be classified. We decided to analyze only the posts correctly classified in order to limit the size of the dataset. This subset is made of 20.5 million posts in 66GB of data. We should also mention that from these 20.5 million, we will use the subset of English posts when doing content analysis. This subset amounts to 11.5 million posts.

4. Structural Analysis

The first thing we do to analyze this dataset composed by blog entries is to see how these posts are interconnected to each other. This is easily represented as a directed graph where nodes correspond to posts and edges correspond to links between them. There are two ways to represent this graph. One, by writing down a node and the links that go out from it to other posts (we call it out-link graph). The second way is to list a node with all the links that point to it (which we will call in-link graph). They are in fact the same graph but some procedures will be easier to perform in one or the other.

4.1 Statistics about the post graph

Out-Link Graph Here we extract all the links found in the posts content and keep track of them if they point to other posts in the dataset. With this we can recreate all the interconnections between nodes.

- Posts with at least one out-link within our reach: 329.258
- Total number of edges: 464.394
- Top 5 posts by number of outgoing edges: 141, 115, 91, 88, 72.

In-Link Graph The in-link version of the graph is just a different representation of the out-link graph and will be useful to have a first idea of post popularity.

- Posts with at least one in-link: 247.579
- Total number of edges: 464.394
- Top 5 posts by number of incoming edges: 1387, 1194, 717, 696, 464.

A quick observation from looking at the number of nodes and edges is that this graph is extremely sparse. When measured accurately, we realize the number of vertices is 521.800. Therefore, 97.5% of posts in the dataset are isolated. This is probably because most of them talk about personal experiences that concern their writers and a very small group of friends. Nevertheless, it is still surprising to find such a low number of links.

4.2 Cascade Extraction

The first topic we want to study are information cascades in the blogosphere. The extraction algorithm is relatively simple: first we look for all the cascade roots (nodes with in-links but without out-links) and then we follow the in-links recursively (Leskovec 2007). For the moment, we are interested in the general properties of cascades so we will take a look at all of them. One way of doing it is to observe the number of cascades according to their size as shown in Figure 1. We notice this graph follows a power law distribution for most of the data. In addition, there is a suspicious number of outliers on the right side of the plot. It would be interesting to see where they come from and if they can be identified as valid and popular cascades.

4.3 Cascade Classification

It is unexpected to see so many big cascades in the size distribution. That implies many posts triggered a cascading effect and were linked (directly and indirectly) by many other elements. Since the rest of the data fits so well with a power law distribution, we wonder why some posts seem to be so popular. The only way to verify this accurately is to manually inspect those cascades and check whether they are indeed important news or interesting topics. After reviewing a few of such *popular* posts, we confirm that there is, undeniably, something suspicious about them. To really

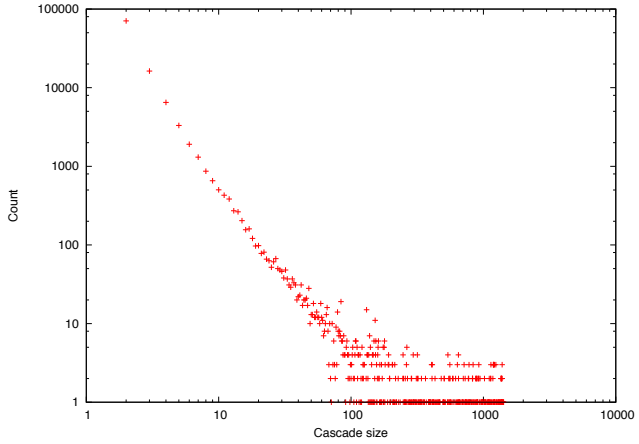


Figure 1: Number of cascades according to size.

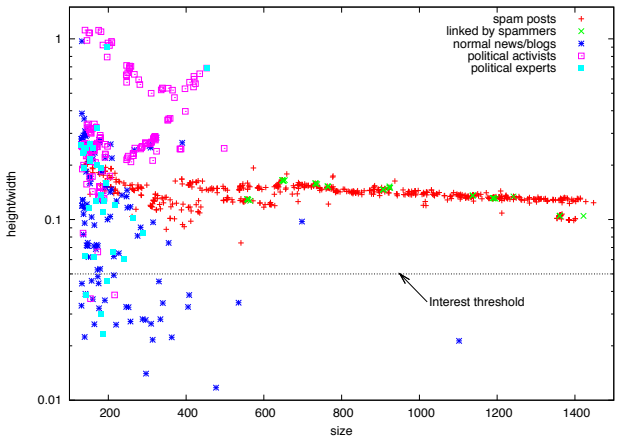
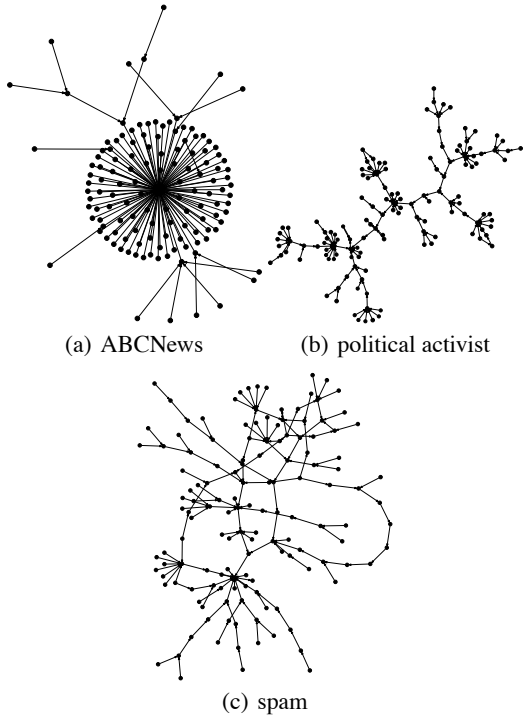


Figure 2: Manual classification for the root nodes of the biggest 1000 cascades.

understand what it is, we decided to review the posts that triggered the biggest 1000 cascades and classify them according to topic. In addition, we wanted to know if there was a correlation between the topic (or its popularity) and the structure of the cascade. To see if this correlation exists, we created a plot where the x axis is the cascade size, and the y axis is a $height/width$ ratio that gives us an idea about the cascade shape. To measure this ratio, cycles are removed and cascades are seen as trees, which enables us to measure their height and width (the tree root is the source node in the cascade). In Figure 2, we can say that there are clear indications of a relationship between the shape of the cascade and its relevance. We cannot really recognize the subject of the cascade (as in politics, technology, etc) but we can see that spammers and *irrelevant* political activists tend to stick together at the upper side of the graph while normal news and *political experts* stay at the bottom. The authors have to point out their lack of knowledge on U.S. politics and for them the line between *activists* (opinionated bloggers) and *experts* (journalists) was sometimes blur.

An easier way to visualize this shape difference is given in Figure 3, in this figure we have three real cascades of comparable size but different shape. On the left, we see a cascade triggered by a normal post coming from ABCNews. Many different bloggers link to it from diverse places and all the nodes we see around the root are directly connected to the source post. On the right, we have the cascade occasioned by a political activist. The beliefs in that blog are strongly defended and opponents are severely criticized. Our intuition tells us that this must be a tight community where the members support ideas of fellow members and will link to them to show agreement. Lastly, we have the cascade at the bottom. Although this one shows a certain similarity to the last one, their topics could not be more different. This cascade is a typical example of a spam blog where the author(s) are probably just trying to improve the site's visibility by increasing the number of links to spam. All the posts in the last cascade belong to the same community and the goal is to lure users to any of those posts.



(a) ABCNews (b) political activist

(c) spam

Figure 3: Cascade shapes for different categories.

4.4 Filtering mechanisms for cascades

Looking at Figure 2 and Figure 3, we realize there is a connection between cascade shape and its popularity. Interesting news tend to trigger star shaped cascades while spam and opinionated posts cause cascades to be more like chains. We can use this fact to establish a simple filtering mechanism to keep important news. We can say we keep all cascades with

$$height/width <= 0.05$$

as shown in Figure 2, but this is an arbitrary threshold and in this case is given only as an illustration. This threshold gives us a recall of 22% and a precision of 94% but our classification is subject to discussion and the very idea of *interesting news* is subjective. Nevertheless, we think the relationship between cascade shape and popularity is significant and should be studied in more detail.

4.5 Temporal Analysis

In addition to the structure, it is interesting to study the time it takes for the cascades to form. The easiest way to see this is to measure the amount of time elapsed between the creation of the root node and the last element of the graph as done in Figure 4.

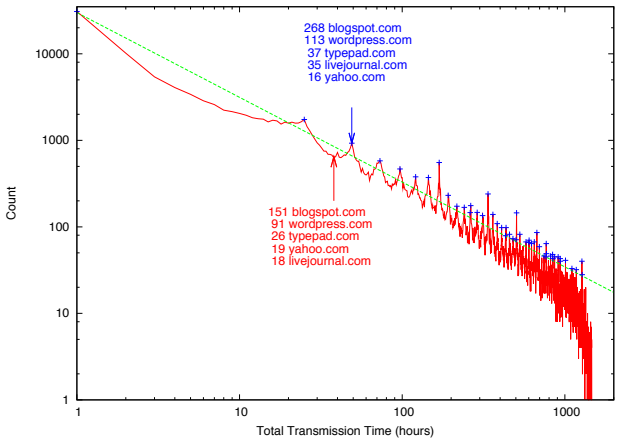


Figure 4: Number of cascades according to their total transmission time (number of posts for the most popular domains are showed for one peak and valley).

There are two interesting observations we can establish from Figure 4. The first one is the cutoff we get after 1500 hours, which is consistent with our expectations since the dataset only covers two months (slightly less than 1500 hours). The second observation, although equally clear from the plot, is harder to understand. Although we expect a power law distribution, we also see cyclical pattern at regular intervals (remember the plot is a log-log scale). To study this phenomenon, we detected all the peaks in the plot and measured the distance between them. The average distance between peaks is 26 hours but a histogram reveals the center at 24 hours and a few outliers that might have arisen from the simplicity of our peak detection algorithm (find the maximums above a manually placed straight line). Explaining this 24 hour pattern is difficult if we are just looking at the data. Something that could give us a hint is knowing where these posts are coming from. With that idea in mind we decided to find the biggest contributors (domains) in terms of number of posts. This is also shown in Figure 4, where we give the number of posts for the top 5 domains in one of the peaks and one of the valleys. Since the behavior seems to be present across domains, our guess is that this observation comes strictly from human behavior. A small but noticeable number of people tend to do things with a given periodicity. This is presumably more prevalent within communities that share the same interests (same country, office hours, etc).

4.6 Conclusions about the structural analysis

In this section we studied the post graph in the blogosphere. We saw how sparse it is and extracted the information cascades within it. The size of such cascades is a basic indicator of public interest since we expect people to link to a post when they are discussing something they like. The problem of this straight approach is that some people realized the same, and constructed artificially big cascades. To help with this problem we proposed to classify cascades according to their shape and concluded that, together with size, the shape of a cascade is a good indication of public interest. We used this knowledge to create a simple filtering mechanism that ignored most of the cascades considered as spam, or which were simply not interesting due to their heavily charged political content. In addition, we analyzed the temporal behavior of cascades and noticed peaks of activities at 24 hour intervals. This is consistent with human behavior where some people do certain things at regular intervals.

5. Content Analysis

Although structural cascades proved to be a good way to discover important news, we believe they still fail to capture some of the posts that talk about events in the real world. While savvy computer users might be more inclined to provide links to sources talking about a new web browser, people giving their opinions about a political campaign do not really have a place to point to. Considering this, it would be very interesting to extract the *cascades* generated by content instead of structure. This is more challenging as it requires us to understand the post body and establish suitable comparisons. It is also subjective because certain people could say two posts are related while others believe they are different. One additional factor is giving an exact shape to the cascade. Since we are looking for similar content, we will end up with *clusters* of posts instead of cascades. This means we will have a bag of posts with relatively similar content but without explicit connections between them.

5.1 Content Clustering

The first step of document clustering is to represent elements in a suitable way for comparison. We decided to use a widely recognize model in the field, the vector space model (Salton 1975), for this representation, and the cosine of the angle between two vectors as the measure for similarity. In this model, a document is given by a vector and each dimension corresponds to a different term. The actual value of this term in the vector could be zero or one if the term is found in the document. It could be the term frequency within the document or it could also be the tf-idf (term frequency-inverse document frequency) which increases the weight of terms that are frequent in the document but rarely found in the whole collection of documents (therefore giving less weight to popular but meaningless words like prepositions).

URL	Title
http://googleblog.blogspot.com/2008/09/fresh-take-on-browser.html	A fresh take on the browser
http://blogoscoped.com/archive/2008-09-01-n47.html	Google Chrome, Google's Browser Project
http://labs.mozilla.com/2008/08/introducing-ubiquity	Introducing Ubiquity
http://blogs.abcnews.com/politicalpunch/2008/09/members-of-frin.html	Members of 'Fringe' Alaskan Independence Party Incorrectly Say Palin Was a Member in 90s...
http://blogs.msdn.com/ie/archive/2008/08/25/ie8-and-privacy.aspx	IE8 and Privacy
http://blogs.msdn.com/ie/archive/2008/06/24/ie8-and-trustworthy-browsing.aspx	IE8 and Trustworthy Browsing
http://blogs.msdn.com/ie/archive/2008/06/10/introducing-ie-emulateie7.aspx	Introducing IE=EmulateIE7
http://foxforum.blogs.foxnews.com	
http://press-releases.techwhack.com/24110-zanox	zanox Marks First Anniversary Since Being Acquired by Axel Springer and Publigroupe
http://blogs.msdn.com/ie/archive/2008/07/02/ie8-security-part-iv-the-xss-filter.aspx	IE8 Security Part IV: The XSS Filter

Table 1: Top 10 posts after filtering (cascade roots).

When processing the dataset to create term vectors, we encounter certain differences with our previous analysis. The clearest one is that since we want to process actual content, we should at least be able to understand it. This forces us to eliminate all non-English posts from the analysis. An additional problem is dealing with user generated content where we encounter an incredible high number of unique words, most of which are not part of the official language. After creating the vector representation for all the English posts (as classified by *spinn3r*), we end up with 11.5 million posts, 833 million words processed and 3.5 million unique words. All this after ignoring case sensitivity, removing stop words and doing stemming.

In Figure 5, we see the distribution of processed words. On the x axis we have the number of times a given word appeared on the dataset and on the y axis the number of words for each count. After noticing the high number of words that only appear a few times, we decided to keep terms that are present at least 5 times in the dataset. With this simplification we decreased the number of unique words from 3.5 million to 750 thousand, which is a considerable reduction in the dimensionality of the space model.

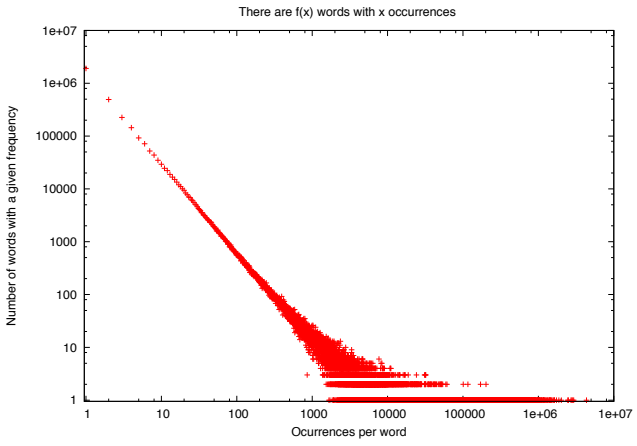


Figure 5: Notice the high number of words encountered only once.

5.2 Online Post Clustering

The next step after representing posts in the space model is to compare them and somehow group them according to their topic. One naive way to do this is to create a similarity matrix, comparing each post to all the others. Since this requires $O(n^2)$ computations, it becomes computationally unfeasible if we have 11 million elements. The complexity given this large scale prevents us from using traditional hierarchical algorithms based on the similarity matrix. Other algorithms of similar (or worse) complexity are also unsuitable, as well as methods were the number of clusters must be known in advance. Such reasons encourage us to look in another direction.

A well known problem in the field of approximation algorithms is the *bin packing problem* (Vazirani 2001). Although this problem is NP-hard, there are heuristics that give relatively good results using only a fraction of the time it would take to find the optimal solution. The statement of the problem in its simplest version is quite straightforward: if there is a series of items of different sizes, how many bins (of a fixed sized) do we need in order to fit all elements?. The optimal solution is found by trying all possible combination of items in the bins. A much faster (but approximate) solution is reached with the following algorithm:

```
foreach item do
    foreach bin do
        if item fits into the bin then
            add it to the bin;
            continue with the next item;
        end
    end
    if item does not fit into any bin then
        create a new bin and put it there;
    end
end
```

The above algorithm is called *first fit* since we leave the item in the first bin where it fits. It is easy to prove that it requires at most twice as many bins as the optimal solution, and it is easy to see that every element is only processed once which is a good indicator of its complexity. We should also mention that there are two flavors of the bin packing problem: offline and online. In the offline version, we have all the elements before starting (we could sort them before fitting for instance). In the online version, items come one by one and we are forced to leave them in bins before processing the next element. The above algorithm is a good way to approach the online version of the problem.

If we apply the same idea to our dataset, we could think of putting similar posts into the same bin. In such a way,

we would end up with clusters of posts. At the end of this clustering we would just look at the biggest clusters, and in theory, we should find the topics that caused the biggest spur in the blogosphere. Our pseudo algorithm would then be exactly like the one given above but replacing the word item by *post*, bin by *cluster* and instead of saying *an item fits into a bin* we will say *a post belongs to a cluster*

The only difference lies between fitting an item in a bin and knowing whether a post belongs to a cluster. The simplest way is to calculate the distance from the post to the cluster centroid in vector space and say they are similar if the distance is less than a certain threshold. This threshold is our only fixed parameter so far and will have a big impact in the running time of the algorithm and in the quality of clusters found. If we require two posts to be very similar in order to put them in the same cluster we will end up with a bigger number of clusters. That means the next post will have to try one additional cluster to see if it fits anywhere. If we go to an extreme and say posts have to be equal in order to be in the same cluster, we will end up doing $O(n^2)$ comparisons again. The problem is that, in fact, posts tend to be quite different in the blogosphere, almost in the same way they were isolated when we were analyzing the post graph in section 4.1.

5.3 Parallel Online Post Clustering

But that same fact of isolation is an advantage for us. What if we can ignore all the posts without a real connection between them and keep only those that talk about common topics?. It turns out that is exactly what we need to do in order to detect important news. It would, of course, be possible if we do the whole clustering and then remove cluster of size one, but how could we remove them before doing the actual processing?. Another critical observation to highlight is that: if entries are ordered by date, we can assume posts related to big news will be relatively close to each other, or we can at least assume they will not be uniformly distributed. If this assumption proves correct, we could do a partial localized clustering and then drop the smallest clusters, all this without missing too many relevant posts. The steps in our analysis would be:

- Partition the dataset in small blocks (ordered by time)
- Perform online post clustering for each block (in parallel)
- Drop the smallest clusters from every block
- Merge clusters in adjacent blocks
- Continue merging hierarchically all the way up until having just one block

An optional step is to remove the smallest cluster on every merge to reduce the computational cost. We removed clusters of size one after the first level, and clusters of size two after the seventh. In Figure 6, we have a visualization of this merging. The elements at the bottom represent the initial cluster blocks and every level represents the cluster merging of two adjacent blocks.

Once we have finished the procedure we can observe the size distribution for clusters in Figure 7. We spot two special points at the beginning, those two points correspond

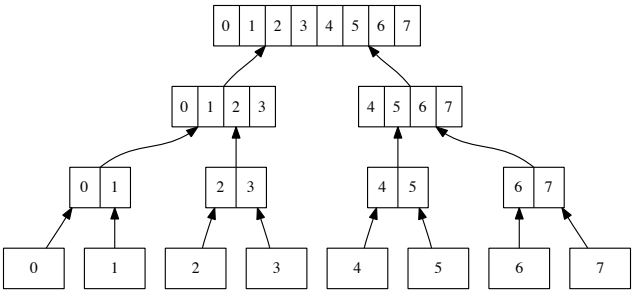


Figure 6: Merging procedure for cluster blocks.

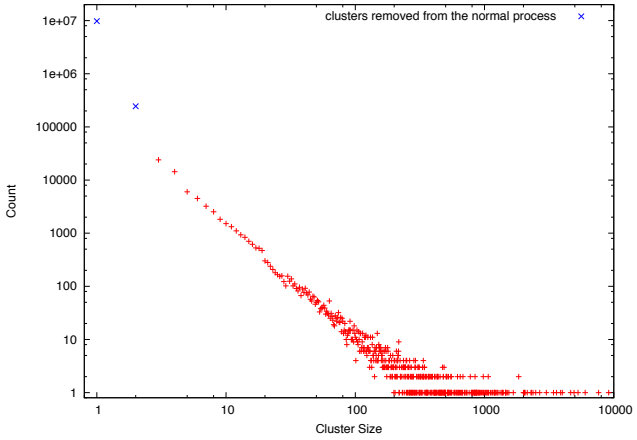


Figure 7: Number of clusters given by size.

to clusters that were pruned during the process. We also remark a power law distribution for the cluster size in the same way we observed it for the cascade size in Figure 1. As we mentioned before, cluster size is directly related to our measure of similarity. If we relax the similarity requirement we would obtain bigger clusters, although such a fact should not change the kind of distribution.

5.4 Cluster Classification

In contrast to the shapes observed in Figure 1, clusters are just bags of posts without any relationship between them. On the other hand, we learned from section 4.3 that variety is a good indication of interest. In that case, diversity was shown as star-like shaped cascades with references coming from many different places, but that is not the only way to see diversification. For clusters, we can measure this diversity by calculating the number of domains or subdomains contained in the post URLs. If all posts in the cluster come from different domains, it means the post is interesting for the general public. We can visualize this diversity in Figure 8 where the elements close to the $f(x) = x$ line are the ones with the highest diversity. We will limit ourselves to subdomains in the future since it gives us a better indication.

At this point we would think clusters close to $f(x) = x$ in

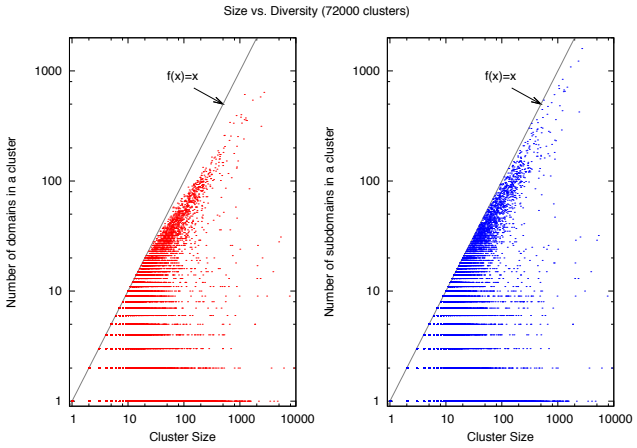


Figure 8: Diversity within a cluster (different domains and subdomains)

Figure 8 are more interesting than others, but can we be sure about this? To get a deeper insight about interesting clusters according to size and variety, we decided to manually classify the biggest 1000 of them by topic. We can see this categorization in Figure 9, where we arranged clusters by main categories. We notice that indeed *real* news tend to be close to $f(x) = x$ while advertising and other kinds of spam are much lower. It is interesting to see how some proper companies contribute to the lower part of the graph. Although they generate many posts concerning their news, they get grouped into few clusters indicating high redundancy. They also come in their totality from a single sub domain, which basically tells us that although there are thousands of posts about a topic, there is only one person talking about it. That is probably the closest we can get to the definition of spam.

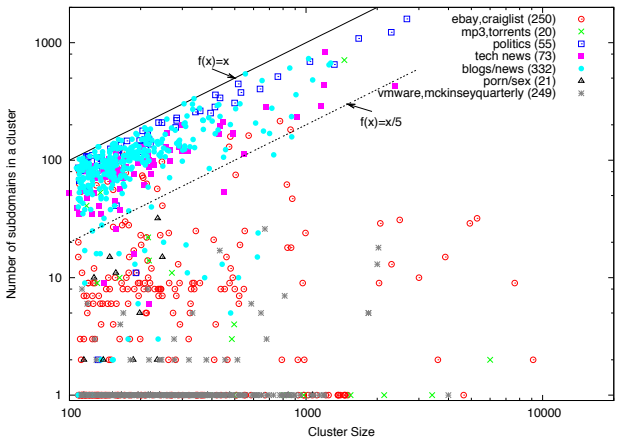


Figure 9: Classification of the biggest 1000 clusters

5.5 Filtering mechanisms for post clustering

Based on Figure 9, we can propose simple strategies to keep only the news that interest the general public. Just observing the graph we can conclude that we would like to keep all clusters that are close to the $f(x) = x$ line. We could then say that we are going to keep all clusters that fall between $f(x) = x$ and $f(x) = x/5$ exactly as plotted in the figure. This surprisingly simple technique gives us decent result with a recall of 88% and a precision of 94%. This is clearly not a definite solution but serves to illustrate the underlying concept of diversity.

5.6 Cluster Visualization

We can analyze the content in the clusters to better understand what they represent. Every cluster is portrayed by its centroid, which is the addition of term vectors for all posts. This centroid can be seen as a list of weighted words and is better understood if we use a graphical representation. In Figure 10, we see a visualization for the biggest eight clusters after filtering. For every cluster, we also give the title of the post that was the closest to the center of the cluster. This central post is a good candidate to represent the whole cluster and manual inspection showed the post body contained a good description of the topic. We should also notice that by comparing Figure 10 and Table 1 we see how the two approaches of the paper give different results, the first approach based on cascades captured mostly technological news while the second approach detected news of more relevance in the real world.

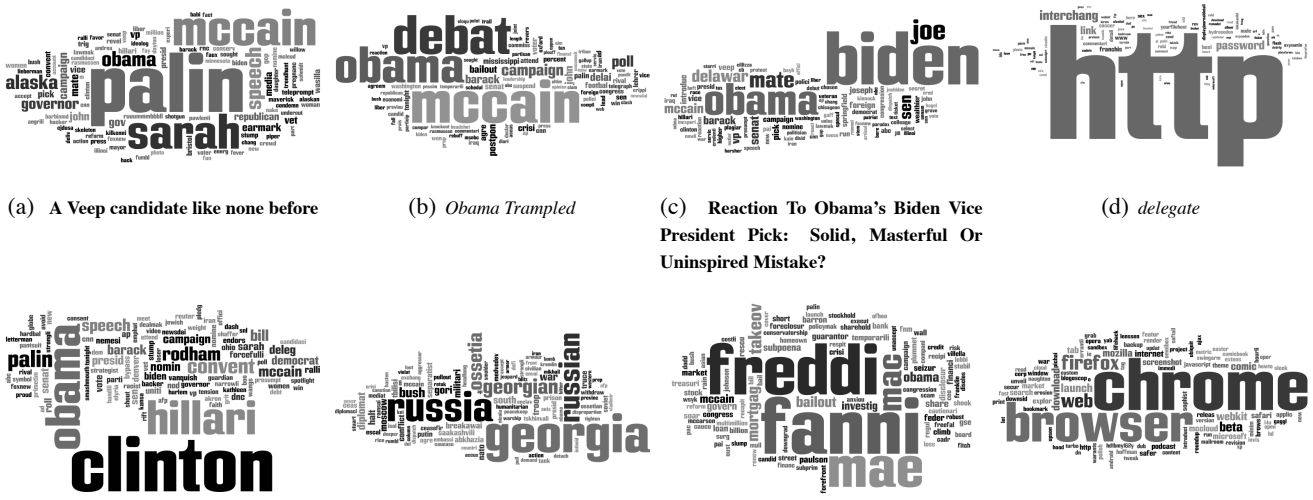
From Figure 10, cluster *(d)* caught our attention since it clearly does not represent an event. After reviewing the posts in this cluster, we realized they were all a list of links. That fact explained the importance given to the word *http* and this can be easily solved in future experiments by adding common internet words to the list of stop words. We leave the cluster in the results for completeness and note that it is the only hollow cluster in the top ten according to Figure 9.

5.7 Conclusions about post clustering

We presented an algorithm to cluster posts according to content. This algorithm is scalable when subjected to pruning and can be performed in parallel at any given level of the merging process (although a formal analysis is needed to measure its exact complexity). It is also incremental as once we have a given set of clusters we can use them to directly merge new posts or new blocks of posts. Apart from the algorithm itself, we measured the diversity within clusters as given by the number of domains and sub-domains as compared to the total number of posts. Based on this measure of diversity, we were able to identify news that were interesting to the general public and filter out posts that were clear efforts of advertising.

6. Final Remarks

This paper focused on two differing and yet complementary approaches to news detection in the blogosphere. The dataset comprised millions of post published during August and September 2008. These posts were first seen as a graph



(a) **A Veep candidate like none before**

(b) *Obama Trampled*

(c) **Reaction To Obama's Biden Vice President Pick: Solid, Masterful Or Uninspired Mistake?**

(d) *delegate*

(e) **Quotes From Around Yon Blogosphere**

(f) *HOW WORLD WARS GET STARTED: Georgian Crackdown On Pro-Moscow Separtists Prompts Massive Retaliation From Russian Army; Bush Plays Volleyball*

(g) **THE POLITICS OF BLACKNESS: Fannie Mae and Freddie Mac: Do you want the truth or the Kool-Aid?**

(h) *Re: Google's own browser... Chrome*

Figure 10: Word clouds for the top eight clusters after filtering, plus the title of the cluster medoid (clouds created at www.wordle.net).

where their structural properties gave good indications about the kind of interest some topics trigger from readers. We then used such properties to differentiate important news from opinionated articles or simple spam. In the second part of the paper, we ignored the structure completely and focused on content. We were able to cluster a large number of posts by topic, and by analyzing the diversity within each cluster, we filtered most of the irrelevant information. When we compare results from both approaches, we notice a strong technological inclination if cascades are used, and more down to earth topics when content is processed. This could come from the fact that tech news usually originate from the internet and that tech bloggers are more prone to use web elements like links, while other bloggers talk about news from the real world with no particular online source. It would be interesting to pursue further research in the differences between both approaches or in how to bring them together, as well as a more exhaustive use of the cluster algorithm to get a better grasp of its precision. Another good research direction would be the temporal analysis of clusters or even finding their structure based on content similarity. That would allow us to have better parameters to measure diversity. Finally, we could use the same kind of analysis on different time frames to perform outbreak detection on short periods and to get a list of world-changing events by focusing on longer intervals.

7. Acknowledgments

This research was done as part of an internship performed at NEC C&C Innovation Research Laboratories. The visiting student warmly thanks the people there for their unmeasurable kindness. He also thanks the other interns for profound and meaningful conversations about research and one thousand other things in life.

References

L. A. Adamic and N. Glance. 2004. The political blogosphere and the 2004 u.s. election: divided they blog. *In the Proc. of the 3rd intl. workshop on Link discovery*.

E. Adar, L. Zhang, L. A. Adamic, and R. M. Lukose. 2004. Implicit Structure and the Dynamics of Blogspace. *WWW2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*.

K. Burton, A. Java, and I. Soboroff. 2009. The ICWSM 2009 Spinn3r Dataset. In the *Proc. of the 3rd Intl. Conf. on Weblogs and Social Media*.

M. Cha, J. A. Navarro, H. Haddadi. 2009. Flash Floods and Ripples: The Spread of Media Content through the Blogosphere. In the *Proc. of the Intl. Conf. on Weblogs and Social Media Data Challenge Workshop*.

D. Gruhl, R. V. Guha, D. Liben-Nowell, A. Tomkins. 2004. Information diffusion through blogspace. *Intl. World Wide Web Conf.*

J. Leskovec, A. Singh, J. Kleinberg. 2006. Patterns of Influence in a Recommendation Network. *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*.

J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, M. Hurst. 2007. Cascading behavior in Large Blog Graphs. *SIAM Intl. Conf. on Data Mining*.

J. Leskovec, L. Backstrom, J. Kleinberg. 2009. Meme-tracking and the Dynamics of the News Cycle. *SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*.

G. Salton , A. Wong , C. S. Yang. 1975 A vector space model for automatic indexing. *Communications of the ACM, p.613-620.*

V. Vazirani. 2001. Approximation Algorithms. *Springer Verlag*.