This paper presents a case study of 'Terms-of-Service; Didn't Read', a social machine to curate, analyse, and rate website terms and privacy policies. We examine the relationships between its human contributors and machine counterparts to determine community structure and information flow.

H.1.2 [**User/Machine Systems**]: Human information processing; G.2.3 [**Discrete Mathematics**]: Applications

Human Factors, Measurement, Design, Theory

Social Machines; User Agreements; Terms of Service; Privacy; Legal Informatics; network science

Social machines are new combinations of human and machine activity that engage in complex activities and solve problems which were previously difficult or costly for humans and machines to do alone [**?**].An interesting application of social machines is to the curation, analysis and rating of legal documents and advice. Contracts, terms-of-service, privacy policies, and licenses all serve important functions in a range of online and offline interactions. However, they incur significant costs on interacting parties as they are often written in complex legalistic language (sometimes referred to as 'legalese') which is time-consuming and difficult for individuals to understand without formal training. In addition, despite advances in automatic parsing techniques made in semantic processing and legal informatics [**?**, **?**], computational processes alone may be insufficient. Instead, social machines aimed at handling these tasks and functions are emerging. These systems aim to capture the activity of human actors who write, develop, read, modify, sign, or abide by such texts, and aggregate that activity to produce new content and services.

We used an existing classificatory framework for social machines to search for and identify systems in this area [**?**]. For the purposes of this paper, a broader analysis of the domain of social machines for parsing legalese has been discarded. We focus here on just one case study, the 'Terms of Service; Didn't Read' project, a platform for analysing and rating website terms and privacy policies.

The ToS;DR platform has around 500 users, who communicate primarily through an open mailing list. Its stated aim is to 'fix the biggest lie on the internet' - namely, the statement that 'I have read and agreed to the terms'. Participants identify, discuss, and annotate clauses in terms-of-service and privacy policies, rating them as either 'good', 'bad', or 'neutral'. This activity generates the raw data which drives the service – as they explain it, 'every thread on the mailing list is a data point'. The number of good, bad and neutral points are tallied to produce an overall rating of a policy. The ratings serve data via an API to other services including a browser plugin.

In order to keep track of changing policies, an automated web crawler called ToSBACK regularly crawls an index of websites and notifies participants of any changes to the policies which may need to be reviewed. Any participant can add a website to the index, which is a list of XPath addresses. Unlike some other social machines we surveyed, this automated 'bot' is treated, according to the ontology of the service, just like any human user, i.e. a node in the mailing list which provides raw data for review.

The ToS;DR platform provides a case study of how legalese text can be parsed through a social machine. The system receives an input of 'raw legalese', analyses its parts in terms of 'good' and 'bad', and outputs an overall score. An interesting aspect of this process is the community structure and the way information flows within it. In order to explore this we used the Girvan and Newman (GN) algorithm for identifying communities in complex networks [**?**]. We ran the same analysis on the data before and after the introduction of the ToSBACK crawler, in order to test how the introduction of a computational actor affected community structure.

We built two graphs G_1 and G_2 corresponding to the mailing list archive before and after the ToSBack bot began con-

tributing. The vertex sets of both graphs are made up of contributors to the mailing list. In total, 2,345 contributions were analysed, from an active user base of between 28-35. An edge was placed between two vertices if the relevant contributors responded, or were responded to by another contributor. These edges were then weighted by the number of interactions between the relevant contributors. Finally we applied the GN algorithm to both graphs.

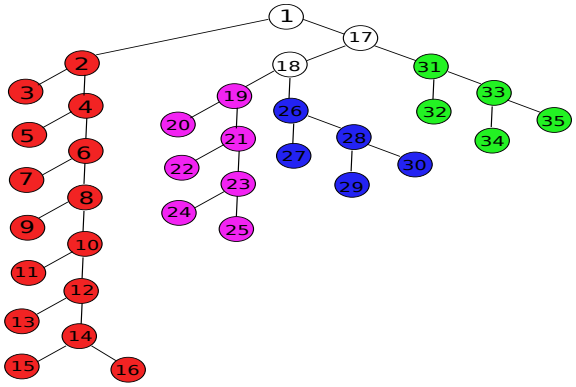

Figure 1: The tree, T_1, built via the GN algorithm that corresponds to the mailing list prior to the ToSBack bot contributions.

Tree T_1 (Figure 1 was built via the GN algorithm from the mailing list *prior* to the introduction of the ToSBack bot, corresponds to a nested hierarchy of communities. The vertex labeled 1 represents the entire mailing list and every interaction between contributors, whereas the vertices 2 and 17 represent sub-communities that we will call $C2$ and $C17$ respectively. Since 2 is adjacent to 3 and 4 there also exist sub-communities of $C2$ that we denote $C3$ and $C4$. One can deduce from the GN algorithm that leaf vertices of T_1 correspond to a single contributor, thus $C3$ is a community of precisely 1 person. On the other hand $C4$ indicates a much larger community – ($C2$ without one contributor).

In general a subtree of the tree associated to a network using the GN algorithm that is "close " to a line indicates a hub and spoke community. We have highlighted four line-like subtrees in T_1; these subtrees can be recognised as the areas of discussion initiated by or frequently involving particular members.

We can formalise the notion of "line-like" subtrees by analysing the *symmetry* of a tree. One can measure how symmetric a tree is by calculating the number of permutations, $|\mathrm{Aut}(T)|$, of the vertices (of that graph) that preserve adjacent vertices [**?**]. We call the set of such permutations the automorphism group of T. In general a low $|\mathrm{Aut}(T)|$ coincides with the presence of one or several long line-like subtrees. We calculated that $|\mathrm{Aut}(T_1)| = 16$ and $|\mathrm{Aut}(T_2)| = 24$. This shows a marked increase in the cardinality of the automorphism group and therefore increased symmetry which is further entrenched by normalisation of $\mathrm{Aut}(T)$.

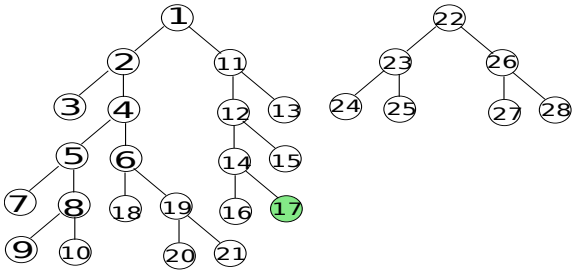

Figure 2: The tree, T_2, built via the GN algorithm that corresponds to the mailing list after the ToSBack bot contributions began. Vertex 17 [green] indicates the bot.

We see two distinct community structures before and after the introduction of the ToSBack bot. Notice the asymmetry of the very long branch (red in T_1), which represents one particular user's tendency to interact with a disproportionate number of contributors. In contrast, after the introduction of the bot, the network is more balanced with no one information hub.

One explanation for this is suggested by research into how structures naturally become optimised. Self-similarity in natural structures like rivers ensures that these structures develop in an optimal way [**?**]. Guimera *et. al.* show that this principle is also true for communities; they tend to self-organise to form an optimal structure [**?**].

One way of measuring self-similarity in a graph is to find the order of the symmetry group or group of automorphisms of that graph. The increasing symmetry in T_2 may therefore be an indication of self-organisation for more efficient dissemination of information.

Finally, one particular user began interacting heavily with the bot accounting for the overwhelming majority of interactions with the bot (86%). During this period, the user's importance in the network grew, triggering more reponses than in the previous period (while her total contributions remained stable). A possible explanation for this is that this user found a new role after the bot arrived; she went from being a producer of content, to become a filter between the new content generated by the bot and other (human) participants.

By applying the GN algorithm to archived interactions between both human and machine actors within a social machine, this paper suggests novel techniques for measuring and analysing several aspects of social machines. This method can reveal how the introduction of a computational actor affects community structure in a social machine. more balanced trees have better more efficient method of disseminating information, spokes and more integrated. looks as if it has made the org more efficient.

Research Councils UK Digital Economy Programme, Web Science DTC, Uni. Southampton, EP/G036926/1. Hugo Roy ToS;DR project lead.

Let G be a simple weighted graph with edge and vertex sets $E(G)$ and $V(G)$ respectively. The importance of an individual edge $e_{ij} \in E(G)$ is commonly calculated in terms of *edge betweenness centrality* which is defined as follows:

$$\beta(e_{ij}) = \sum_{u \neq v \in V(G)} \frac{\sigma_{uv}(e_{ij})}{\sigma_{uv}}$$

where σ_{uv} is the number of shortest paths from vertex u to v and $\sigma_{uv}(e_{ij})$ is the number of those shortest paths that pass through edge e_{ij}.

In order to determine community structure we will apply the Girvan and Newman (GN) algorithm [**?**] which associates a binary, rooted tree, T, with a simple weighted graph G as follows:

(i) The root of T is assigned to be the whole graph G.

(ii) The edge, e_{ij}, with the highest betweeness centrality in G is determined.

(iii) Edge e_{ij} is removed from G.

(iv) If step (iii) disconnects G then connect two vertices to the root of T (these vertices correspond to the connected components of G).

(v) Iterate until there are no remaining edges in G.

It has been shown [**?**] that the degree of cohesion in a network can be detected via the GN algorithm and it has been used to identify communities in structures as diverse as scientific collaboration networks, food webs and e-mail networks [**?**, **?**].